National Research Council Canada
Conseil national de recherches Canada

Division of Electrical Engineering
Division de génie électrique

ERB-980

Recognition of Three-dimensional Objects based on the Cross-angle Transform

R. Oka, T. Kasvand, and M. Rioux

April 1986

Canada

NRCC No. 25661

86-06448/

Copyright 1986 by
National Research Council of Canada

Permission is granted to quote short excerpts and to reproduce figures and tables from this report, provided that the source of such material is fully acknowledged.

Copyright 1986 par
Conseil national de recherches du Canada

Il est permis de citer de courts extraits et de reproduire des figures ou tableaux du présent rapport, à condition d'en identifier clairement la source.

Additional copies are available free of charge from:

Editorial Office, Room 301
Division of Electrical Engineering
National Research Council of Canada
Ottawa, Ontario, Canada
K1A 0R8

Des exemplaires supplémentaires gratuits peuvent être obtenus à l'adresse suivante:

Bureau des publications, Pièce 301
Division de génie électrique
Conseil national de recherches du Canada
Ottawa (Ontario) Canada
K1A 0R8

RECOGNITION OF THREE-DIMENSIONAL OBJECTS BASED ON THE CROSS-ANGLE TRANSFORM

R. Oka, T. Kasvand, and M. Rioux

ERB-980

NRCC No. 25661

APRIL 1986

TABLE OF CONTENTS

Page

ABSTRACT vii
RÉSUMÉ viii
1. INTRODUCTION 1
2. OUTLINE OF 3-D OBJECT RECOGNITION 2
3. FACET EXTRACTION 3
3.1 The Range Image 3
3.2 Pre-processing 4
4. THE CROSS-ANGLE TRANSFORM 12
4.1 The Two Degrees of Freedom 14
4.2 Ambiguity Elimination 15
5. REPRESENTATION OF 3-D STRUCTURE 18
5.1 Structure Representation From a Single View 18
5.2 Structure Representation From Multiple Views 18
6. PRIMARY MATCHING FOR DETERMINATION OF A REPRESENTATIVE FACET PAIR 20
7. VERIFICATION ALGORITHM FOR A RESULT OF PRIMARY MATCHING 22
8. EXPERIMENTAL RESULTS OF 3-D OBJECTS RECOGNITION 24
8.1 An Example of the Parameters Used for Verification 30
8.2 An Example of Repeated Verification Process 35
8.3 Analysis of Verification Process 38
8.4 The Weaker Condition for Determining Matched Facet Pairs . . 41
9. DIRECTIONS FOR FUTURE RESEARCH 44
9.1 Mathematical Analysis of Ambiguity Reduction 44
9.2 Verification by the Use of Connectivity and Shape of Facets 44
9.3 Occlusion Problems 47
10. CONCLUSIONS 47
ACKNOWLEDGEMENTS 48
REFERENCES 48
APPENDIX A: RELATION BETWEEN A GRADIENT VECTOR AND A NORMAL VECTOR . 51
APPENDIX B: ROTATION EQUALIZATION OF AN OBJECT USING TWO COMMON FACETS OF DIFFERENT VIEWS 53
APPENDIX C: PROGRAMS FOR 3-D OBJECT RECOGNITION 57

LIST OF FIGURES

1 Outline of the 3-D recognition system 2
2 The 3-D camera which gives the elevation image Z(X, Y) 4
3 (a) The geometry of the Z(X, Y) image for pre-processing. (b) The direction cosines. (c) The convolution matrix w 5
4 (a) The dynamically thresholded, binarized, and labelled decision space $I_h(s, n)$ for 5° intervals, s = 1, 2, ..., 72,

Page

n = 1, 2, ..., 18. (b) A portion of the region labelled image $L_r(i, j)$. (c) A portion of the edge image labelling of edges and corners 9
5 Description of 3-D objects by facets 13
6 The cross-angles between facets 13
7 Ambiguity of a single A_i for reconstructing of the original object 14
8 An example of the reduction of ambiguity using two A_i's 15
9 The ambiguity reduction concerning the normal vector of the jth factor 16
10 (a) The intersection part of two conical surfaces as one open curve. (b) The intersection part of two conical surfaces as one closed curve. (c) The intersection part of two conical surfaces as two closed curves, and the intersection part of three conical surfaces yielding eight points 17
11 The union of two facet representations 19
12 The experimental result for making a union of two facet descriptions 20
13 The three-dimensional data for learning the model descriptions by using the Cross-Angle Transform 25
14 The three-dimensional data to be recognized 25
15 The facet segmentation of the three-dimensional data of Fig. 13 . 26
16 The facet segmentation of the three-dimensional data of Fig. 14 . 27
17 The sub-scene numbers and the facet numbers of each sub-scene of Fig. 15 28
18 The sub-scene numbers and the facet numbers of each sub-scene of Fig. 16 29
19 The quantizations for the cross-angles α and β 30
20 (a) The model numbered 3; (b) scene blobs numbered 4, 9, and 6 . 42
21 The definition of consistency loop 46
A-1 The notations of the normal vector (a, b, c), the slant angle, and the tilt angle 51
B-1 The schematic explanation of the transform for obtaining the union of the first and the second angle facet sets 53
B-2 The notations of the normal vector (a, b, c) and the angles ψ and θ 55
B-3 The notations of (B_x, B_y, B_z), (B''_x, B''_y, B''_z), ψ_1, ψ_2, θ_1, θ_2 . . 55

LIST OF TABLES

1 The number of points included in the intersected part of three conical surfaces generated by three β's 18
2 Distance matrix between the three largest facets of each model and the facets of all the blobs of input 31

Page

3 The pairs of the model facet and the scene facet which give the minimum distances among the blobs in Table 2 32
4 An example of the configuration of the "children" facet pairs in order to verify the correspondence of Table 2 for recognizing the model number 1 33
5 Generation of the "children" facets for three candidate representatiive pairs for recognizing the model 9 36
6 The data used for repeated verification 39
7 The connectivity matrices 45
8 The brother facet pairs of the father (8, 10) composed of 8th facet of the model 1 and 10th facet of the scene 5 46
9 The new brother facet pairs for Table 8 by eliminating false facet pairs using connectivity characteristics 46

ABSTRACT

A new method, called the Cross-Angle Transform, is proposed for both describing and matching a three-dimensional structure. The Cross-Angle Transform is applied to a description of facets constituting a 3-D object, where each of the facets has a normal vector, an area, and the location of its center. The Cross-Angle Transform is defined as a transform from a facet to a set of angle-pairs by using two kinds of cross-angles between normal vectors or orientation vectors from a facet to all of the facets. A set of angle-pairs has shift and rotation invariant characteristics for any 3-D object. A single set of angle-pairs cannot reconstruct completely the original 3-D object because of two degrees of freedom caused by the two kinds of cross-angles. However, the ambiguity can be reduced if we use several sets of angle-pairs. Matching between models and a scene including objects is carried out by (1) calculating the distances between facets of models and facets of a scene using two-dimensional histograms with respect to sets of angle-pairs, (2) determining a representative facet pair based on the distances, (3) producing the possible corresponding pairs with respect to a representative pair, and (4) verifying the representative pair. This report shows some recognition results which lead us to believe that the method is applicable to real scenes including 3-D objects.

RÉSUMÉ

On propose une nouvelle méthode appelée "transformée par angles croisés" pour décrire et apparier une structure tridimensionnelle. La transformée par angles croisés est appliquée à une description de facettes constituant un objet tridimensionnel, où chaque facette est définie par un vecteur normal, une superficie et l'emplacement de son centre. La tranformée par angles croisés transforme une facette et un ensemble de paires d'angles en utilisant deux types d'angles croisés entre les vecteurs normaux ou les vecteurs d'orientation d'une facette par rapport à toutes les facettes. Un ensemble de paires d'angles ne peut reconstruire complétement l'objet tridimensionnel à cause des deux degrés de liberté résultant des deux types d'angles croisés. Cependant, on peut réduire l'ambiguité en utilisant plusieurs ensembles de paires d'angles. La correspondance entre des modèles et une scène comprenant des objets s'obtient de la façon suivante: (1) on calcule les distances entre les facettes des modèles et les facettes d'une scène en utilisant des histogrammes bidimensionnels par rapport à des ensembles de paires d'angles, (2) on détermine une facette représentative en se basant sur les distances, (3) on produit les paires correspondantes possible en se référant à une paire représentative, (4) on verifie la paire représentative. La présent rapport donne quelques résultats d'essais qui nous font penser que la méthode peut s'appliquer à des scènes comportant des objets tridimensionnels.

RECOGNITION OF THREE-DIMENSIONAL OBJECTS BASED ON THE CROSS-ANGLE TRANSFORM

R. Oka, T. Kasvand, and M. Rioux

1. INTRODUCTION

The "intelligence" component of a machine vision system consists of descriptions and matching of shapes. The objective of this report is to propose a new method called Cross-Angle Transform. The method is used for both describing and matching a 3-D structure. The method is applied to a description of facets obtained by the method described originally in another paper (1).

Few matching algorithms have been proposed which are both applicable to real 3-D objects and are also well formalized. It is difficult to recover the 3-D scale and the rotation transforms, and there are processing errors, such as missing information or splitting of facets, which are usual in the 3-D world. The matching methods, generalized Hough transform by Ballard (2) and extended Gaussian image by Bajcsy (3), are well known for solving the above problems, while relaxation and maximum clique detection methods are used as general tools (4, 5). In general, the field of machine vision is exceedingly active and also very diverse. Some examples of range finder camera developments are given in refs. 6-8, image prepocessing in refs. 9-15, and object recognition techniques in refs. 16-24.

Contrary to these methods, which seem to be either very general or quite specific, the method proposed in this paper uses simpler and more intrinsic characteristics of the available images for 3-D object description by facets. That is, two cross-angles are defined between normal vectors or orientation vectors from one facet to another in a 3-D description by facets. For each facet, a set of angle-pairs is made by using two kinds of cross-angles in order to create a structural description of an object. The cross-angles are shift and rotation invariant in the 3-D world and also are available to use for the purposes of describing and matching of 3-D objects. The description of the 3-D structure is defined by the set of cross-angles itself. Matching between models and a scene including objects is carried out by (1) calculating the distances between facets of models and facets of a scene using two-dimensional histograms, (2) determining a representative facet pair, (3) producing all possible corresponding facet pairs with respect to a representative pair, and (4) verifying of the representative pair.

In this report we will describe (1) the details of a facet extraction algorithm from range data, (2) the concept and the algorithm of the

Cross-Angle Transform, (3) some experimental results of the 3-D object recognition experiment, and (4) the future directions of the research based on the Cross-Angle Transform.

2. OUTLINE OF 3-D OBJECT RECOGNITION

A recognition system for 3-D objects based on the facet extraction (1) and the Cross-Angle Transform is realized as shown in Fig. 1.

The input scene to be recognized is usually separated into a set of blobs by the facet description algorithm (1). In the following, the term **a scene** means **a blob (object cluster) in the scene to be recognized.**

The main body of the recognition system consists of the following five parts:

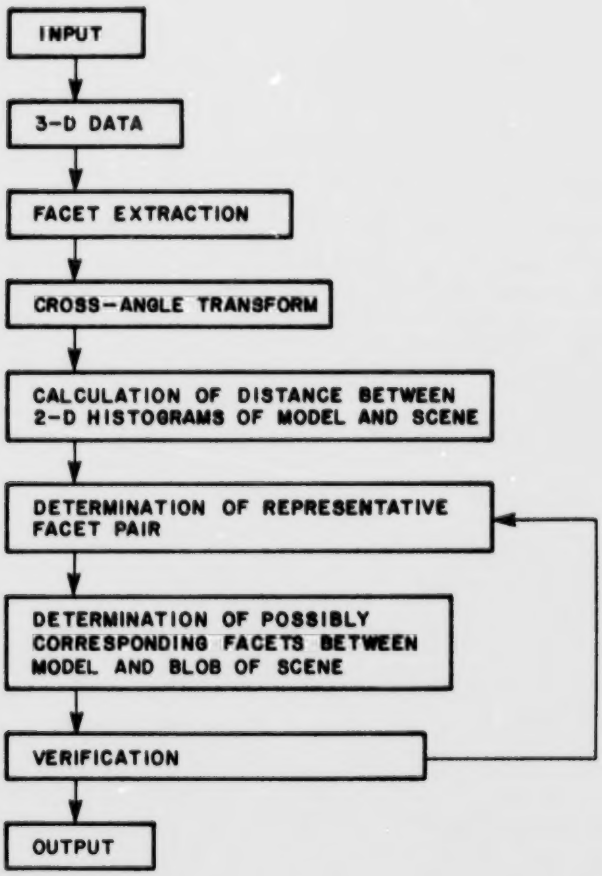


Figure 1. Outline of the 3-D recognition system.

(1) facet extraction from a range data

(2) structure description of a 3-D object using the Cross-Angle Transform

(3) determination of a representative facet pair giving the minimum distance between facets of models and facets of a scene

(4) generation of all of possible matched facet pairs between a model and a scene with respect to a representative facet pair

(5) verification of the representative facet pair by eliminating false facet pairs in the facet pairs obtained in (4)

3. FACET EXTRACTION

3.1 The Range Image

In the present case the 3-D scenes are polyhedra of the "blocks or origami world" type (25) as well as cylinders, spheres, and other relatively simple curved objects. Only opaque smooth surfaces of various sizes and in various orientations are allowed. Otherwise, no constraints have been placed on the number of objects, their shape, position, orientation, or the overlap among the objects in the scene. The scene is scanned with a laser range finder type scanner which provides a two-dimensional image of distances Z(X, Y) from a zero reference plane (X, Y), orthogonal to the line of sight from the scanner. The distance measurement Z at a particular location (X, Y) is called a pixel. A pixel Z(X, Y) gives the elevation Z at location (X, Y) on the ground plane, as seen from above, where X and Y are the coordinates of the reference plane. Actually, the 3-D camera makes measurements from a point above the scene, but the readings are the actual Z-values obtained by using a mechanical synchronization of two scanners (see Fig. 3.1). Such an arrangement provides a reference plane that can be set at any distance along the Z axis. This is done by geometrical adjustments of the camera head. Details of the camera may be found in ref. 26. Since the camera is stationary, all points in the scene are not visible from the camera, which results in certain "blind regions" and "cliffs" in the image caused by the viewing geometry. Additional noise is introduced by multiple reflections of the scanning beam. Isometrically displayed data are shown in Section 8 (Figs. 13 and 14.)

To readers familiar with traditional image processing, the "gray levels" in the image are now the elevation values Z(i, j), i = 1, 2, ...,

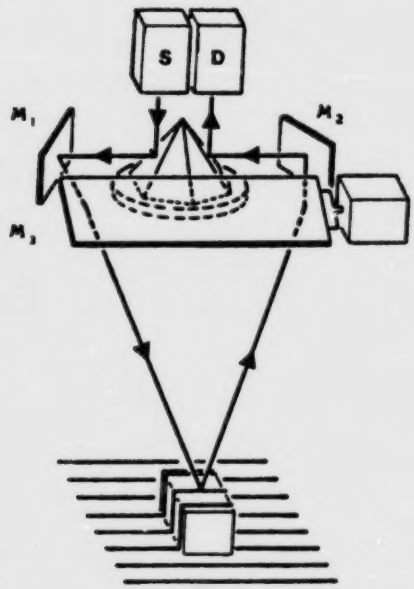


Figure 2. The 3-D camera which gives the elevation image Z(X, Y). M1, M2, and M3 are mirrors, S and D are the laser light source and detector. The scanning in the X-direction is done by the rotating multifacetted mirror and in Y-direction by M3.

I_m, j = 1, 2, ..., J_m, where i and j are the coordinates of the digital image matrix. The relationships between the (i, j) coordinates of the image matrix and the (X, Y) coordinates of the reference plane are X = i and Y = -j. The location (X_0, Y_0, Z_0) of the camera relative to the ground plane is at $X_0 = I_m/2$, $Y_0 = -J_m/2$, $Z_0 = Z(I_m/2, J_m/2)$. The spatial resolution in the present case is approximately 1 mm/pixel, and the image size is 128 by 128 (I_m = 128, J_m = 128).

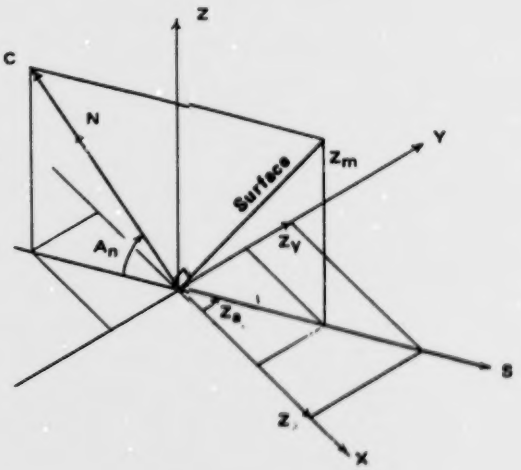
3.2 Pre-processing

The goal of pre-processing is to segment the image Z(X, Y) of the 3-D scene into individual surfaces or facets, to detect the edges and corners, and to present this information and the inter-relationships between the surfaces to other programs for subsequent analysis (1). In the present case several series of computations are carried out to extract as much information as feasible from the image Z(X, Y), even though all of it may not be needed by subsequent programs in a given case. Each processing stage extracts and condenses the information into "larger wholes," resulting at the end in a description for each surface, edge, and corner as, an individual entity, and their inter-relationships.

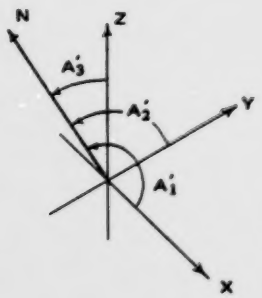
3.2.1 *Local Information from 3 by 3 Areas*

These processing steps use the Z(X, Y) data directly or after some smoothing (median filtering over an n by n local area) to reduce noise. The local area used is 3 by 3 pixels, or n = 3, to minimize noise but also to keep the loss of spatial resolution to a minimum. The geometry and some of the quantities computed are shown in Fig. 3a and b. The 3 X 3 local area operator is shown in Fig. 3c. The basic mathematics may be found in textbooks on geometry and image processing.

(a)



(b)



$A_1 = \cos(A_1')$
$A_2 = \cos(A_2')$
$A_3 = \cos(A_3')$

(c)

$$\begin{matrix} w_4 & w_3 & w_2 \\ w_5 & w_0 & w_1 \\ w_6 & w_7 & w_8 \end{matrix}$$

Figure 3. (a) The geometry of the Z(X, Y) image for pre-processing. The symbols are described in the text. (b) The direction cosines. (c) The convolution matrix w.

The Gradients

The X and Y gradients Z_x and Z_y of Z(X, Y) are $Z_x = dZ/dX$ and $Z_y = dZ/dY$. These are computed at each pixel ($i = 1, 2, \ldots, I_m$; $j = 1, 2, \ldots, J_m$) from the elevation values Z(i, j) by convolving Z(i, j) with a 3 × 3 operator, where $w_1 = w_2 = w_8 = 1$, $w_0 = w_2 = w_7 = 0$, $w_4 = w_5 = w_6 = -1$, and $w_2 = w_3 = w_4 = 1$, $w_0 = w_1 = w_5 = 0$, $w_6 = w_7 = w_8 = -1$. Usually, the Z_x and Z_y values are smoothed once ($w_k = 1$, $k = 0, 1, \ldots, 8$). The magnitude Z_m of the gradient of the elevation values is $Z_m = (Z_x^2 + Z_y^2)^{1/2}$. The angle of the gradient of the elevation values Z_a indicates the direction of greatest rise in the elevation values. $Z_a = \text{Arctg}\ (Z_y, Z_x)$, where Arctg is the inverse tangent operation.

Ordinary and Jump Edges

In a range image there are two kinds of edges, namely, the "ordinary edges" and the so-called "jump edges." At an ordinary edge one surface joins, meets, or overlaps another surface. At a jump edge there is a sudden and significant change in the Z value across the edge, i.e., it is a "cliff edge." At a jump edge it is fairly likely, but of course not certain, that one surface obscures the other surface. Both the magnitude of the gradient Z_m and the change of the gradient magnitude dZ_m/ds are large at a jump edge. The distance element ds is in the direction of the angle Z_a of the gradient Z_m. Consequently, jump edges occur when $Z_j = (Z_m) * (dZ_m/ds) > T$, where T is a threshold. A value for T may be found from the histogram of Z_j. The "ordinary edges," however, cannot be detected reliably from thresholding of Z_m since the value of Z_m varies over too wide a range. The local maximum of Z_m or of $Z_e = (Z_m - b * \left|dZ_m/ds\right|)$ may be tried, where b is a constant, but there is no guarantee that the resultant edge is continuous. Here both edge types have been computed differently, to guarantee continuous edges.

Local Surface Normals

The surface normals in a local n by n area may be found by least-squares surface fitting (27). For n = 3, the least-squares method and the gradient computations above are identical in form. Alternatively, the normal to the surface N is orthogonal to the gradient magnitude Z_m. Consequently, the angle of the normal N projected onto the XY plane is $Z_a + 180°$, or $A_s = \text{Arctg}\ (-Z_y, -Z_x)$ (Fig. 3a). Considering a distance ds = 1 on the XY plane along the direction S of the gradient, and since the gradient is orthogonal to the surface normal N, the elevation angle $A_n = \text{Arctg}\ (1, Z_m)$. The direction cosines of the normal N, A_1, A_2, and A_3, with respect to the X, Y, and Z axes, are $A_1 = -Z_x/C$, $A_2 = -Z_y/C$, and $A_3 = 1/C$, where $C = (Z_x^2 + Z_y^2 + 1)^{1/2}$. Estimates of the local surface curvatures K_0 and K_{90} are obtained from the change of direction of the local normal vectors.

3.2.2 Grouping of Local Information

The next stage of pre-processing consists of grouping or clustering the local results in order to detect the surfaces (facets), the edges (vertices), and the corners. This consists of three major steps: first, to find a mapping M which transforms selected localized parameters into a new space H where the desired surface types form clusters; second, to detect and mark the clusters; and third, to use the cluster information to locate the surfaces, edges, and corners in the image Z(i, j).

Clustering

A familiar example of mapping and clustering is the Hough transform, which transforms lines in the image space into points in the transform space. This could be used on the detected edges, if their shapes are known. In the so-called "gradient space," H_g, the axes are gradients Z_x and Z_y; in the "angle space," H_a, the axes are the orientation A_s and the tilt A_n; in the "surface curvature space," H_k, the axes are the two curvatures K_0 and K_{90}, and so on. Actually, higher dimensional spaces, such as $H_a * H_k$, offer even better possibilities. The H_a and H_k spaces are preferred here since their range is limited (i.e., the axes are in the ranges 0 to 90, 0 to 360, and -180 to 180°).

The gradients $Z_x(i, j)$ and $Z_y(i, j)$ or the angles $A_s(i, j)$ and $A_n(i, j)$ of the normal vector are theoretically constant for any plane surface. For smooth curved surfaces these values vary continuously as a function of the position on the surface. In the H_a space each plane surface becomes a point, a cylinder or cone becomes a curve, while a spherical surface becomes a point in the H_k space. However, due to various geometric aberrations, surface irregularities, and noise effects, only clusters are formed in these spaces. In the present case the $A_s(i, j)$ and $A_n(i, j)$ values are accumulated in a two-dimensional histogram or decision space $H_a(s, n)$, where the s-axis corresponds to $A_s(i, j)$ and the n-axis to $A_n(i, j)$. The axes have been quantized to $s = 1, 2, \ldots, 72$ intervals and $n = 1, 2, \ldots, 18$ intervals, corresponding to 5° resolution of $A_s(i, j)$ and $A_n(i, j)$. The plane surfaces at various angles in Z(i, j) will correspond to tight clusters in $H_a(s, n)$, cylindrical and conical surfaces form elongated clusters, but spherical surfaces do not cluster.

Cluster Analysis

It should be noted that the clustering procedure in the decision space determines the surfaces detected in the image space. Each cluster is detected and labelled uniquely, such that the label of the cluster can be used later in the image plane Z(i, j) for detection of the corresponding surfaces.

There are many clustering procedures (28). In the present case the clusters in $H_a(s, n)$ are separated by dynamic thresholding and the separate clusters are labelled and expanded. The resultant labelled decision space $L_h(s, n)$ for the image in Fig. 11 is shown in Fig. 4a. The horizontal surfaces are treated as a separate case, since $Z(i, j)$ is not used, but now, for example, a horizontal cylindrical surface on the XY plane is split into three regions, i.e., the horizontal portion and the two sides.

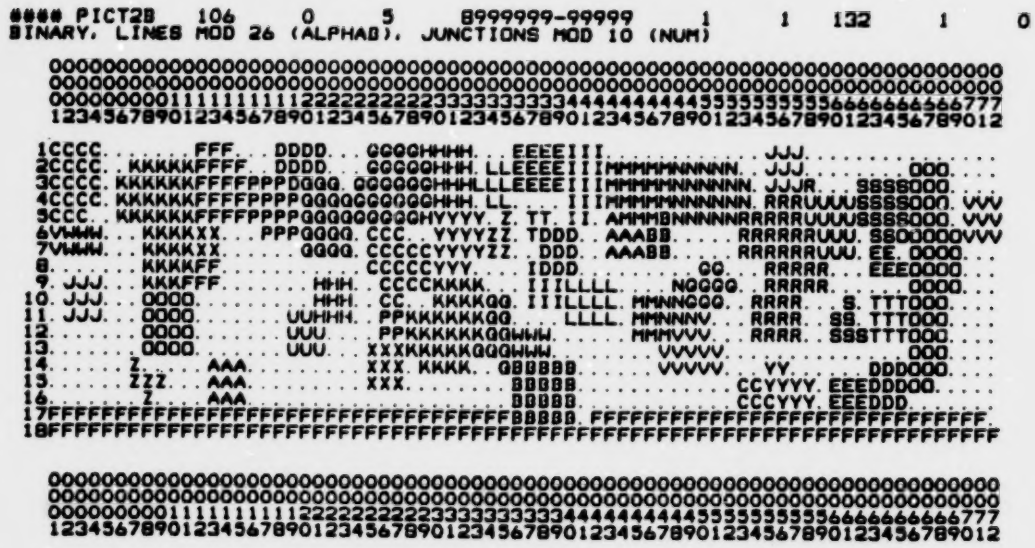
Back Projection, Facets, Edges, and Corners

The labelled clusters in $L_h(s, n)$ correspond to surfaces in the image $Z(i, j)$. The cluster labels in $L_h(s, n)$ are "projected back" into the image plane by using the A_s and A_n values. This gives an image of cluster labels $L_h'(i, j)$. The clusters representing the individual surfaces in $L_h'(i, j)$ are separated by edge detection in a labelled image. In brief, given a 3 × 3 four-connected region in $L_h'(i, j)$, if the label of the central pixel differs from the label of any of its four neighbours, then the central pixel is zeroed. This results in a "cleaned" back-projected image $L_h(i, j)$, which in the present case has been masked by the thresholded $Z(i, j)$ image $B_z(i, j)$, in order to emphasize the object regions. The edges are detected by pixel-to-pixel Boolean operations between binary images derived from the already available images, the edges are thinned, giving image $B_t(i, j)$, and the areas enclosed by the thinned edges are labelled, resulting in image $L_r(i, j)$ (see Fig. 4b). Each edge and corner in $B_t(i, j)$ is uniquely labelled (see Fig. 4c), and the edges are approximated by piece-wise linear curves. The processing of $B_t(i, j)$ follows the methods outlined in ref. 29, resulting in several additional registered images, each carrying specific information about the original image.

A study of the region labelled image $L_r(i, j)$ reveals that at the boundaries between larger surfaces several small regions may have been created which do not exist in the original image. Such regions are also created at "jump edges." Due to differencing and smoothing, any region which is smaller than about 5 × 5 pixels or thinner than about 5 pixels should be considered unreliable. Further study is required to determine whether these errors can be eliminated during pre-processing. Alternatively, these small regions may simply be rejected by the subsequent programs, after correcting for connectivity between the remaining regions.

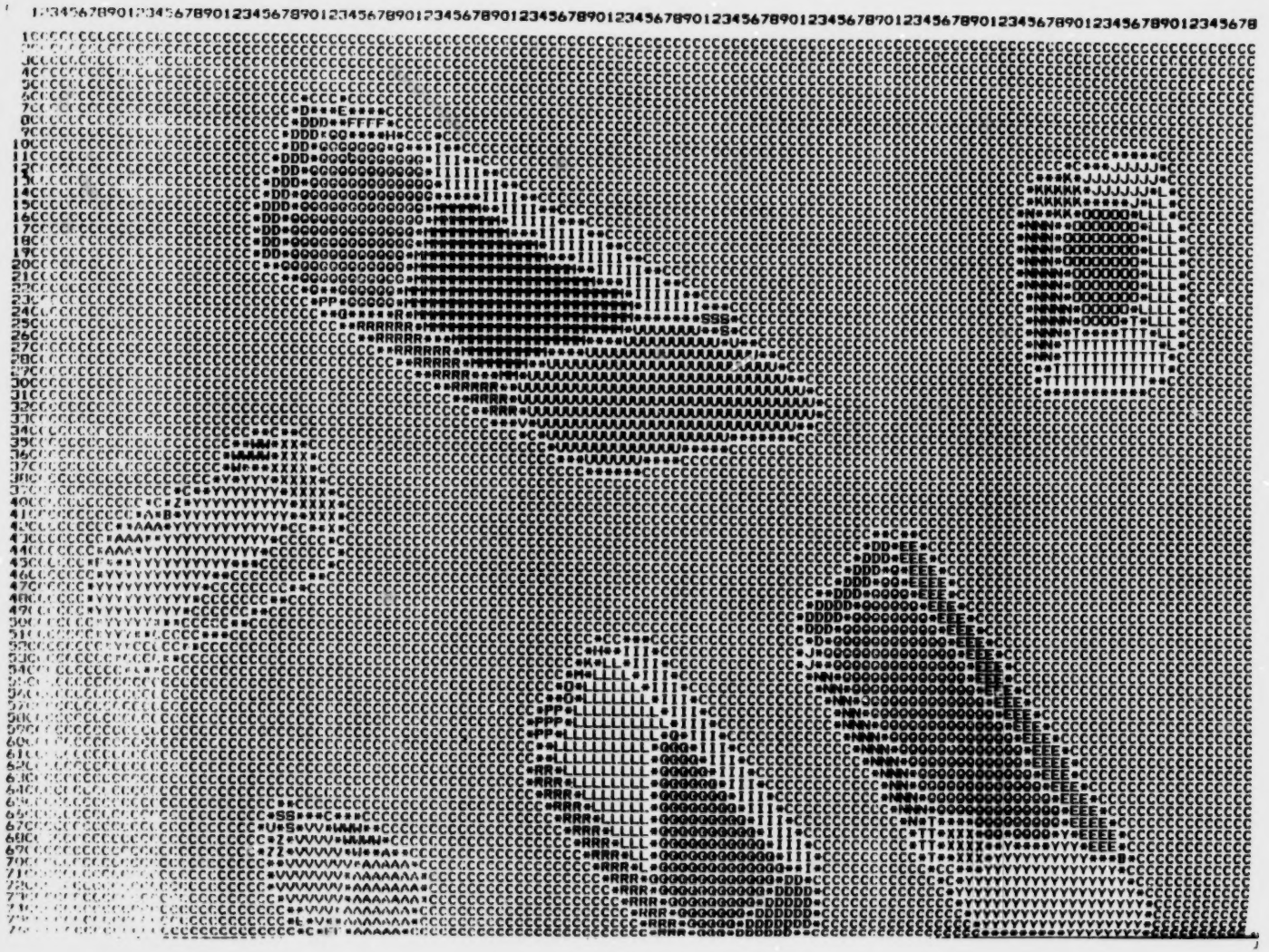
3.2.3 An Elevation Constraint

In the present case the objects are on a flat ground plane. This is also the case in many industrial applications where the objects are, say, on a conveyor, a table, or on the floor. Consequently, the elevation image $Z(i, j)$ is thresholded at level Z_0, resulting in a binary image $B_z'(i, j)$,



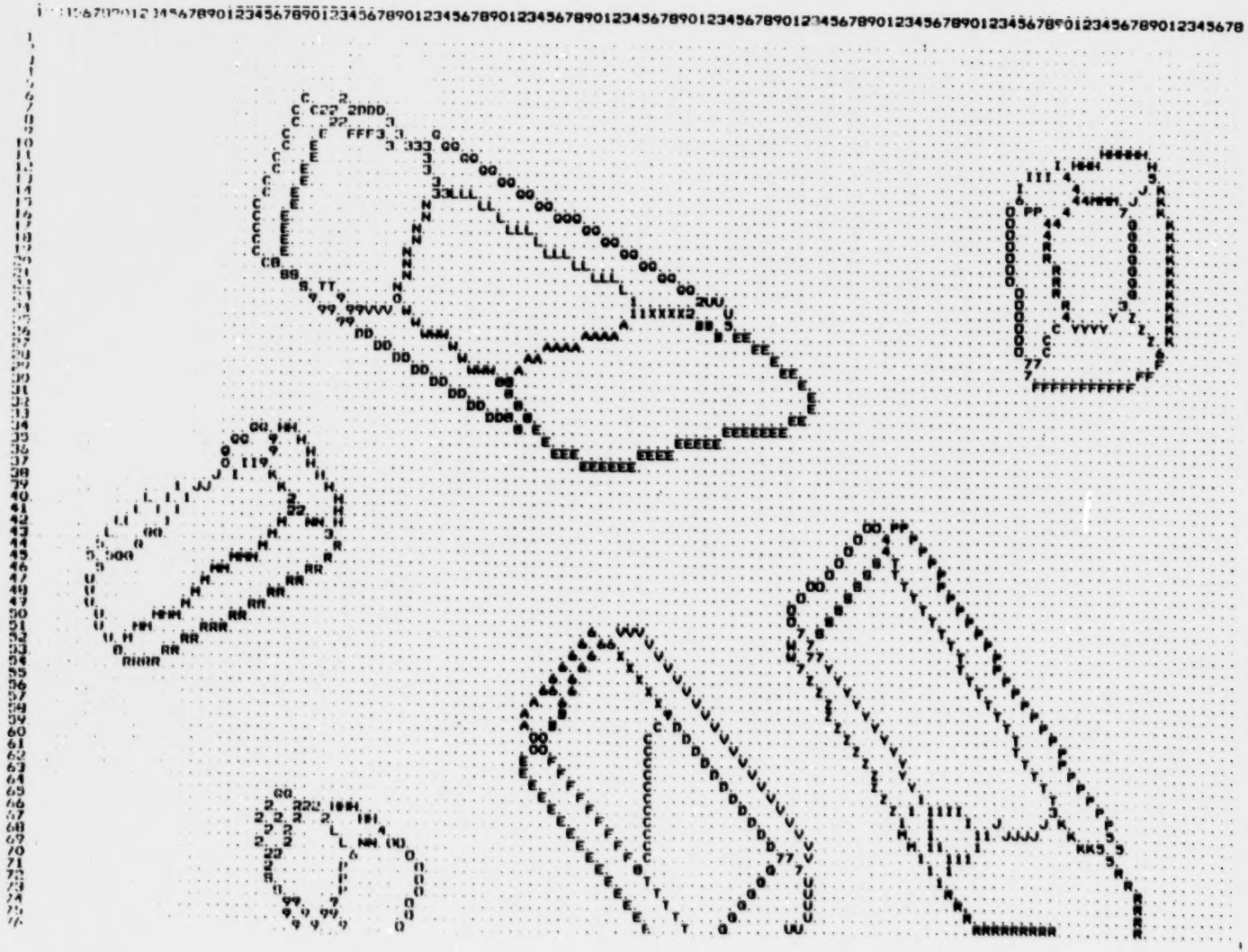
(a)

Figure 4. (a) The dynamically thresholded, binarized, and labelled decision space $L_h(s, n)$ for 5° intervals, s = 1, 2, ..., 72, n = 1, 2, ..., 18. The letters C, D, ... indicate individual surface labels. (b) A portion of the region labelled image $L_r(i, j)$. The letters indicate region labels (facets), the stars (*) indicate boudaries or edges between the regions. (c) A portion of the edge image labelling of edges (the letters C, D, ...) and the corners (the numbers). The corners have been "expanded" to eliminate one-pixel-long edges.



(b)

Fig. 4 (Continued)



(c)

Fig. 4 (Concluded)

i.e., $B'_z(i, j) = 1$ if $Z(i, j) > Z_0$, else $B'_z(i, j) = 0$. $B'_z(i, j)$ is "cleaned," giving $B_z(i, j)$, and labelled for connectivity (four-connected labelling), giving $L_z(i, j)$. The threshold Z_0 may be found from a histogram of $Z(i, j)$ or Z_0 may be available from prior knowledge. The labelled regions in $L_z(i, j)$ represent single objects or object clusters if the objects are touching or overlapping.

3.2.4 Comments

The pre-processing has produced sets of registered images, which contain a variety of information in easily accessible form. Thus, the labels in the region labelled image $L_r(i, j)$ and those in the thresholded and labelled elevation image $L_z(i, j)$ may serve as addresses to data on surfaces within given object clusters. The statistics on surface normals for each individual surface are obtained from $A_s(i, j)$ and $A_n(i, j)$ for each label in $L_r(i, j)$. The connectivity between the labelled regions is determined from $L_r(i, j)$ by using a 3×3 operator centered on the edge pixels. The shape and size of each labelled region or surface, as seen from the view angle of the 3-D camera, are computable from $L_r(i, j)$, and so on. The results may now be assembled in a list or tabular form. Most of the pre-processing steps can be realized as special image processing hardware to speed up the processing.

4. THE CROSS-ANGLE TRANSFORM

Let us assume that an object is described by facets, each of which has a normal vector, an area, and the location in three dimensions as shown in Fig. 5. The Cross-Angle Transform may be regarded as a method of obtaining a representation of the object structure by means of producing a set of angle pairs based on cross-angles between facets. Figure 6 shows the two facets which determine two kinds of cross-angles α, β $(0 \leq \alpha, \beta \leq \pi)$ where angle α is the cross-angle between the normals n_i and n_j, and angle β is the angle between the normal n_i and the orientation vector c. The vector c is the vector from the point $a(x_i, y_i, z_i)$ to $b(x_j, y_j, z_j)$, where (x_i, y_i, z_i), (x_j, y_j, z_j) indicate the center coordinates of the ith and jth facets.

Let α_{ij} and β_{ij} be the cross-angles of the ith facet with respect to the jth facet; then these angles are defined using their normal vectors and one orientation vector as follows. The α_{ij} is defined by:

$$\tan \alpha_{ij} \triangleq \frac{\left(\begin{vmatrix} M_i & N_i \\ M_j & N_j \end{vmatrix}^2 + \begin{vmatrix} N_i & L_i \\ N_j & L_j \end{vmatrix}^2 + \begin{vmatrix} L_i & M_i \\ L_j & M_j \end{vmatrix}^2 \right)^{1/2}}{L_i \cdot \underline{L} + M_i \cdot \underline{M}_j + N_i \cdot \underline{N}}$$

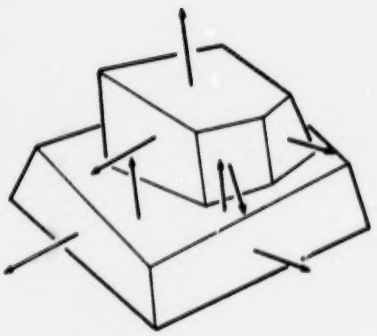


Figure 5. Description of 3-D objects by facets.

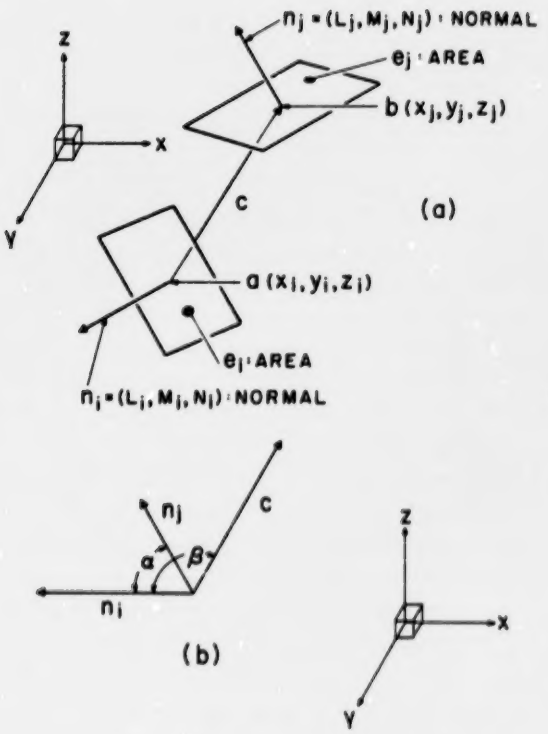


Figure 6. The cross-angles between facets. The α is the cross-angle between n_i and n_j. The β is the cross-angle between n_i and c.

where (M_i, L_i, N_i) and (M_j, L_j, N_j) are the orientation vectors of the ith and jth normals. Using the orientation vector $c = (\underline{L}, \underline{M}, \underline{N})$, the cross-angle β_{ij} is defined by:

$$\tan \beta_{ij} \triangleq \frac{\left(\begin{vmatrix} M_i & N_i \\ \underline{M} & \underline{N} \end{vmatrix}^2 + \begin{vmatrix} N_i & L_i \\ \underline{N} & \underline{L} \end{vmatrix}^2 + \begin{vmatrix} L_i & M_i \\ \underline{L} & \underline{M} \end{vmatrix}^2 \right)^{1/2}}{L_i \cdot \underline{L} + M_i \cdot \underline{M} + N_i \cdot \underline{N}}$$

The Cross-Angle Transform is defined as a transform from each facet in the set of facets constituting a 3-D object to a set of angle pairs, namely,

$$A_i \triangleq \{(\alpha_{ij}, \beta_{ij}): \; j = 1, 2, \ldots, m\}$$

where i indicates a facet of the object, j indicates another facet of the same object (j = i is allowed), and m is the total number of facets of the object.

If an object is composed of m facets, set $A = \{A_i: \ i = 1, 2, \ldots, m\}$ is obtained, which becomes the description of the object.

4.1 The Two Degrees of Freedom

A single A_i cannot reconstruct completely an original object, because there are two degrees of freedom when reconstructing the original object. Let us consider the ith set of pairs, namely A_i, of an object. Let us define the jth facet as an arbitary facet of the same object; then it is impossible to determine uniquely the relation between the ith facet and the jth facet with respect to the relative locations and the relative orientations of the normals. There are two degrees of freedom for the relation. The two degrees of freedom are shown in Fig. 7. One of the degrees of freedom means that a conic surface becomes the possible area on which the center point of the jth facet is located, while the ith facet is located at the vertex of the conic surface. The other degree of freedom is concerned with the freedom of the orientation vector of the jth facet. This freedom indicates another conic surface with the vertex on a point of the former conic surface. This conical surface has a size such that the length of the line from the vertex to a point of the circle is 1. We call this conical surface "surface of **unit cone.**" The possible normal vectors of the jth facet are the vectors from the vertex along the surface of the unit cone.

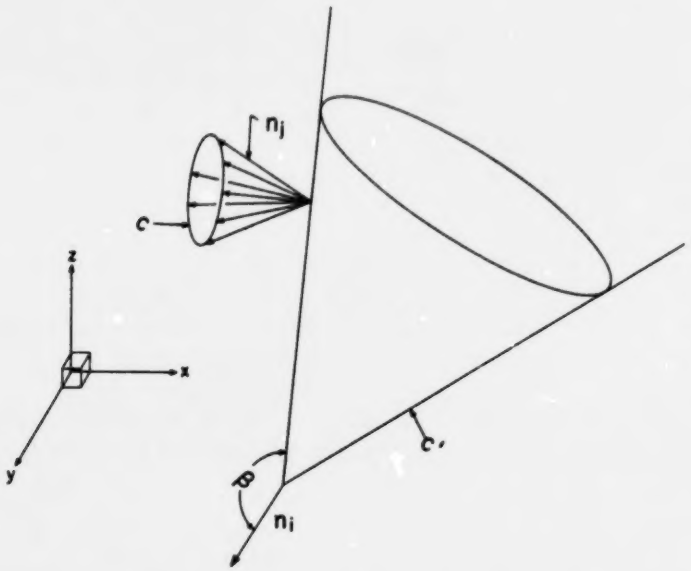


Figure 7. Ambiguity of a single A_i for reconstructing of the original object. Conic surface C/C' is ambiguity caused by α/β angle.

4.2 Ambiguity Elimination

Let us assume that the cross-angles $\alpha_{i,j}$ and $\beta_{i,j}$ are defined for the facet i with respect to an arbitrary facet j. Then the ambiguity caused by the two kinds of freedom concerning $\alpha_{i,j}$, $\beta_{i,j}$ indicates both the uncertain normal vector and the relative location of the facet j with respect to the facet i.

The ambiguity can be reduced if we consider some A_i's. Figure 8 shows an example of the reduction of the ambiguity in using two A_i's. In this case, the ambiguity caused by the α angle can be reduced to two angles as the possible ones, while the location ambiguity caused by the β angle becomes the set of points on the curved and closed line in 3-D space which is the intersection line of the two conical surfaces of $\beta_{i_1,j}$ and $\beta_{i_2,j}$.

Let us consider a general problem of the reduction of the ambiguity concerning all facets consisting of a 3-D object. If the ambiguity is eliminated completely, the structure of a 3-D object is determined uniquely.

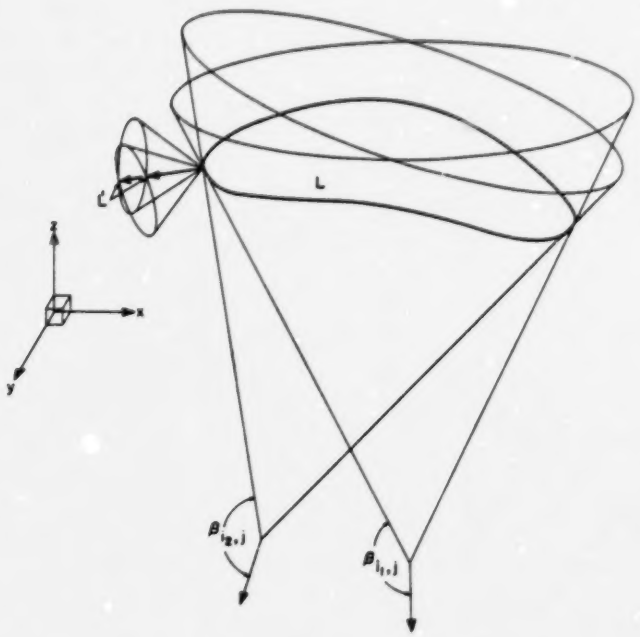


Figure 8. An example of the reduction of ambiguity using two A_i's. The ambiguity for the location of the jth facet is reduced to the bold closed line (L). The ambiguity of the normal of jth facet is reduced to two bold arrows (L').

Let us define three cross-angle pairs by $(\alpha_{i_1,j}, \beta_{i_1,j})$, $(\alpha_{i_2,j}, \beta_{i_2,j})$, and $(\alpha_{i_3,j}, \beta_{i_3,j})$, where i_1, i_2, i_3 indicate three different facets of an object and j indicates an arbitrary facet of the same object.

First, let us consider the reduction of ambiguity of the normal vector of facet j. The unit conical surfaces generated by $\alpha_{i_1,j}$ and $\alpha_{i_2,j}$ are not identical even if both of their vertices are located on a point of the intersection part of the conical surfaces generated by $\beta_{i_1,j}$ and $\beta_{i_2,j}$, since the rotation axes of the conical surfaces generated by $\alpha_{i_1,j}$ and $\alpha_{i_2,j}$ are not always identical, as shown in Fig. 9.

Therefore, their intersection, namely, the ambiguity of the normal, indicates a set of two lines included in both the conical surfaces generated by $\alpha_{i_1,j}$ and $\alpha_{i_2,j}$. If we use three angles, $\alpha_{i_1,j}$, $\alpha_{i_2,j}$, $\alpha_{i_3,j}$, the normal vector is uniquely determined, because the intersection of the three surfaces of the cones becomes a point. The arrow from the vertex of the unit cone to the intersection point becomes the normal vector of the facet j.

Second, let us consider the ambiguity reduction of relative locations between facets, that is, reduction of the intersection part of some cones generated by some β's. It is easy to see that the intersection of two

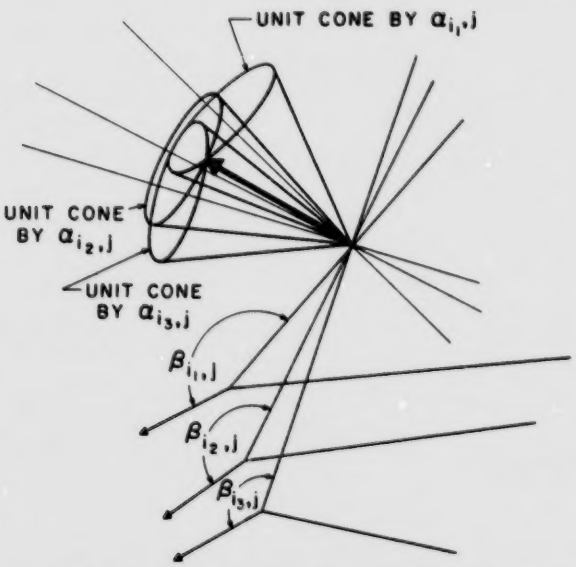


Figure 9. The ambiguity reduction concerning the normal vector of the jth factor. The intersection part of three surfaces of a unit cone is a point.

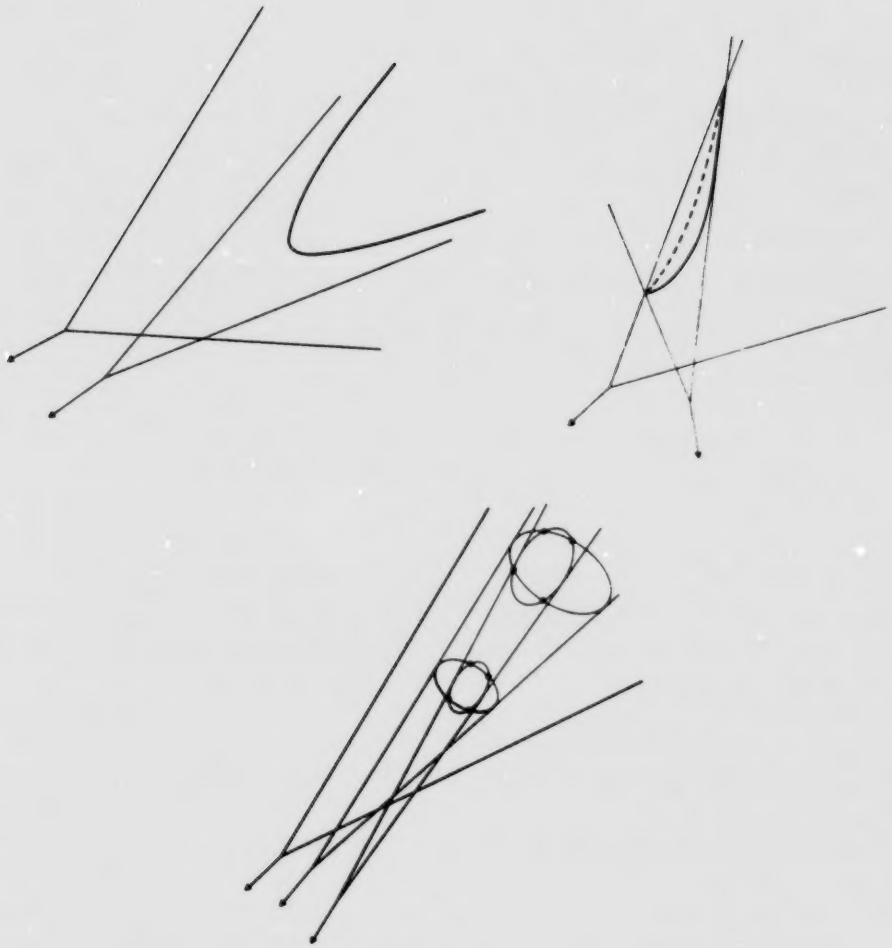


Figure 10. (a) The intersection part of two conical surfaces as one open curve. (b) The intersection part of two conical surfaces as one closed curve. (c) The intersection part of two conical surfaces as two closed curves, and the intersection part of three conical surfaces yielding eight points.

circular cones generated by $\beta_{i_1,j}$ and $\beta_{i_2,j}$ indicates one convex/concave open curve or one/two convex closed curves in 3-D space.

Then we can obtain some points in 3-D space as the maximum intersection part of three conical surfaces as shown in Table 1.

Therefore the intersection part of three cones, C_1, C_2, C_3, generated by $\beta_{i_1,j}$, $\beta_{i_2,j}$, $\beta_{i_3,j}$, becomes a maximum of eight points in 3-D space, if

Table 1. The number of points included in the intersected part of three conical surfaces generated by three β's.

Intersection of 1st and 2nd conical surfaces	One open curve	One closed curve	Two closed curves
Intersection of 1st and 3rd conical surfaces	Intersection part	Intersection part	Intersection part
One open curve	2	2	8
One closed curve	2	2	8
Two closed curves	8	8	8

we select $\beta_{i_1,j}$, $\beta_{i_2,j}$, $\beta_{i_3,j}$ such that there are no cone pairs (C_i, C_j), $i, j = 1, 2, 3$, $i \neq j$, with the same intersection curves. In order to reduce the ambiguity of relative locations completely, we need additional conditions satisfied by three or more β's; these will be discussed in another paper (22).

5. REPRESENTATION OF 3-D STRUCTURE

5.1 Structure Representation From a Single View

In a previous section we assume that all the facets consisting of a 3-D object are obtained before determining a model description by using the Cross-Angle Transform. However, it is obvious that all the facets cannot be obtained by processing the 3-D range data of a single view. Since there are many invisible facets in the range data of a single view, a set of A_i ($i = 1, 2, \ldots, n$) produced by data of a single view using the Cross-Angle Transform is regarded as a part of the complete representation of the 3-D object. This part of the complete representation is useful only for the case where objects have two-dimensional rotations.

5.2 Structure Representation From Multiple Views

A set of facets covering all facets of a 3-D object is required to obtain a complete structure representation of a 3-D object based on the Cross-Angle Transform. The complete structure representation is used for recognizing the objects which have arbitary rotations and overlap each other.

Let us show the algorithm for obtaining the complete structure representation. First of all we will consider making a union of two different sets of facets obtained by the same 3-D object. If these two sets have at least two common and known facets, it is possible to obtain the union of these two facet representations by equalizing the rotation and the shift between these two facet representations. The mathematical method for the equalization is shown in Appendix B. Figure 11 shows the union of two facet representations.

Figure 12 illustrates the experimental result when we make a union of the first facet description (Fig. 12a) and the second (Fig. 12b). In the real data, many facets are missing and there are many split facets when extracting facets from range data.

The complete set of facets is obtained by repeating of the union operation until all the different facets are included. By applying the Cross-Angle Transform to a complete set of facets, we can produce an ideal complete representation of 3-D structure, that is,

$$A = \{A_i: \ i = 1, 2, \ldots, n\}$$

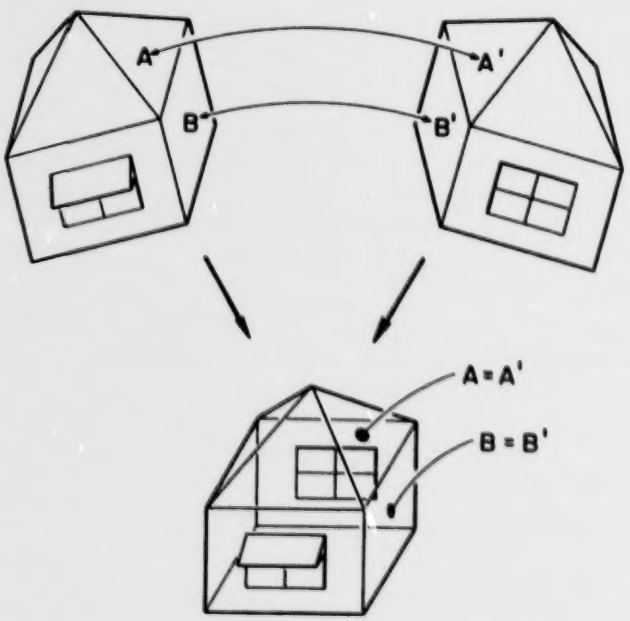


Figure 11. The union of two facet representations.

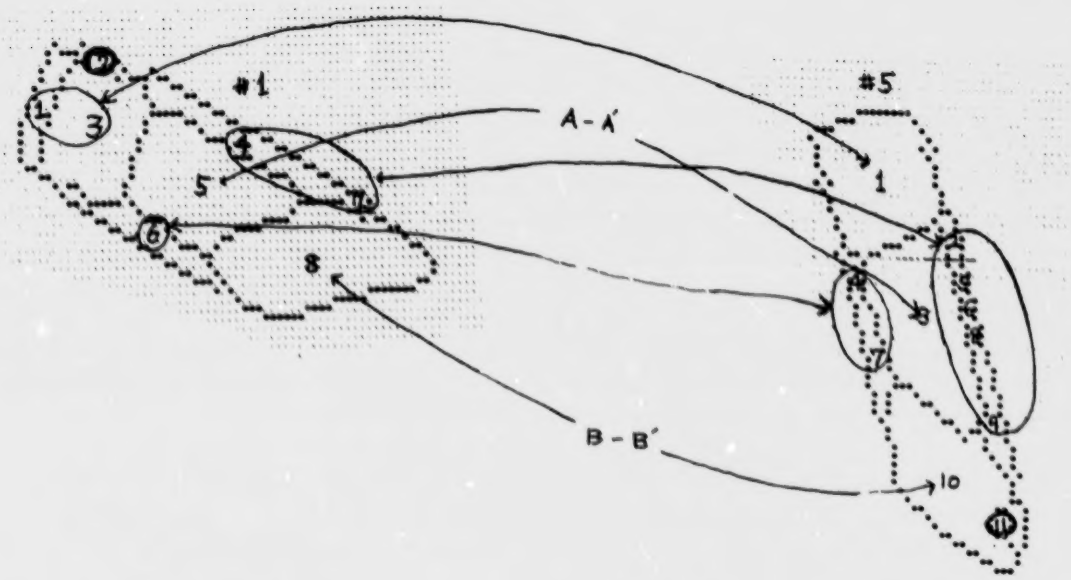


Figure 12. The experimental result for making a union of two facet descriptions.

6. PRIMARY MATCHING FOR DETERMINATION OF A REPRESENTATIVE FACET PAIR

The precise recognition process is divided into two stages, that is, the primary matching and the verification. The purpose of the former stage is to obtain a candidate facet pair between a model and a scene which determines the starting situation of the recognition process in cooperating with the latter procedure, namely, verification. In this section we will show the procedure of the primary matching.

First of all, we will define a two-dimensional histogram with respect to an A_i called the β-histogram. This β-histogram is used to calculate the distance defined between A_i and A_j.

An $\alpha\beta$-histogram for A_i is defined by

$$H_i = \{h_i(\alpha, \beta): \ 0 \leq \alpha, \beta \leq \pi \}$$

$$h_i(\alpha, \beta) \overset{\Delta}{=} \sum_{j=1}^{m} \delta(\alpha - \alpha_{ij}) \cdot \delta(\beta - \beta_{ij}) \cdot e_j$$

where m is the total number of facets of an object, e_j is the area of the jth facet, and $\delta(X) = 1$: $X = 0$, $\delta(X) = 0$: $X \neq 0$ holds.

Even though a single A_i with an $\alpha\beta$-histogram is one element of the representation of a 3-D object, it has the following useful characteristics for 3-D object recognition:

(1) Both an A_i and a H_i are obviously shift- and rotation-free for any 3-D objects because the angles α, β are invariant for shift and rotation in 3-D space.

(2) Even if a facet is split once, the corresponding value on the histogram does not change if the quantized β does not change (facet connectivity-free).

The former makes the recognition system viewer-independent, while the latter leads to robustness.

According to the discussion about the ambiguity of A_i for reconstructing the original object, three A_i's are not enough to determine uniquely the correspondence between the objects of the model and the scene. However, in the real application, sometimes it is enough to use three A_i's for obtaining the correspondence. Meanwhile, it is difficult to identify a correct pair, since we have the following two problems:

(1) Since facets of a 3-D object in a real scene cannot be obtained exactly with respect to the center location, the normal vector, and the area, it is difficult to obtain the complete set of A_i's with $\alpha\beta$-histograms which includes all rotations and other variations. Usually, we can obtain only a subset of it by means of merging sample objects with different views.

(2) The method of determining a correct pair using A_i's and $\alpha\beta$-histograms is not obvious, because it is difficult to define the distance between an A_i of a model and an $\underline{A_i}$ of a scene.

One algorithm for solving these problems is described in the following.

Let us define the distance between $\alpha\beta$-histograms. Let $\{H_i(\alpha, \beta): (\alpha, \beta) \in N_1 \times N_2, 1 \leq i \leq m\}$, $\{h_j(\alpha, \beta): (\alpha, \beta) \in N_1 \times N_2, 1 \leq j \leq n\}$ be the $\alpha\beta$-histograms of the ith facet in a model and jth facet in a scene. Then

$$D_{i,j} \triangleq \sum_{\alpha,\beta} d_{i,j}(\alpha, \beta)$$

where

$$d_{i,j}(\alpha, \beta) \triangleq \begin{cases} \left| H_i(\alpha, \beta) - h_j(\alpha, \beta) \right| : & H_i(\alpha, \beta) = 0 \\ 2 \cdot h_j(\alpha, \beta) : & H_i(\alpha, \beta) \neq 0 \end{cases}$$

It is obvious that $D_{i,j}$ is a kind of distance measure between αβ-histograms, because two αβ-histograms become identical when a facet selected from a model object corresponds with a facet of an object in a scene. For each model, the (i, j) pair whose $D_{i,j}$ gives the smallest value among $\{D_{i,j}: 1 \leq i \leq m, 1 \leq j \leq n\}$ is regarded as a representative pair. For a representative pair, it is possible to produce a set $G_{i,j}$ of possible pairs between facets of a model and facets of a scene, for which angle values of α's (and β's) are identical. Let us define S_i and S_j as the ith facet of a model and the jth facet of a scene, respectively. Then the set $G_{i,j}$ is defined by:

$$G_{i,j} = \{(S_q, S_r) \mid C_1(S_i, S_q) = C_1(S_j, S_r), \; C_2(S_i, S_q) = C_2(S_j, S_r)\}$$

where S_q/S_r is the facet of the object in the model/scene, and $C_1(f_1, f_2)$, $C_2(f_1, f_2)$ are cross-angles α, β between f_1 and f_2 facets, respectively.

A representative pair is called a "father" pair. An element of $G_{i,j}$ is called a "brother of father" pair. A "father" pair is supported by its "brother of father" pairs.

With respect to a set of facet pairs, $G_{i,j}$, all elements of it do not describe correct correspondences of facets between a model and a scene, since a single A_i has ambiguity about facet correspondences. The facet pair composed of the ith facet of a model and the jth facet of a scene is only the best estimate.

7. VERIFICATION ALGORITHM FOR A RESULT OF PRIMARY MATCHING

Thus, we have two verification problems to solve. They are correlated with each other. The one is to verify correctness of a representative pair, namely the (i, j) pair, and the other one is to verify correctness of element pairs of $G_{i,j}$. In this situation there is no logical priority of processing order of verifications. We will propose an algorithm which carries out the two verifications in a combined way.

[Verification Algorithm]

(1) Selecting a representative pair (i, j) (a "father" pair) for each model by detecting the minimum distance among $\{D_{i,j}: 1 \leq i \leq I_m,$

$1 \leq j \leq J_s\}$, where I_m is the facet number of the model m and J_s is the facet number of the scene.

(2) Producing $G_{i,j}$ (a set of "brother of father" pairs) for the representative pair (i, j). The total number of $G_{i,j}$ is defined by r_{ij}. Let us count the number of facets which are included in the model with ith facet, are different from each other, and are included in $G_{i,j}$. The number is defined by $g_{i,j}$. A $g_{i,j}$ can be regarded as a measure representing the strength for supporting the (i, j) pair by other pairs, since, if the (i, j) pair is a correct correspondence, then most elements of $G_{i,j}$ become correct pairs at the same time in spite of including ambiguity, and $g_{i,j}$ becomes nearly equal (exactly equal in the ideal case) to the total number of facets of the model object. Therefore we can use this measure, namely $g_{i,j}$, as the likelihood of matching for the pair (i, j).

(3) Producing "children" G_{i_k,j_k} with respect to an element pair of $G_{i,j}$. The element pair of $G_{i,j}$ is regarded as a new representative pair, where (i_k, j_k) indicates an element (a "brother of father" pair) of $G_{i,j}$.

(4) Counting the number (P_{i_k,j_k}) of facets (of a model) which are different from each other and also are included in G_{i_k,j_k} for each element (i_k, j_k) of $G_{i,j}$. The number P_{i_k,j_k} also indicates the strength of support of the (i_k, j_k) pair by other pairs for the same reason as that discussed in point (2). We will eliminate redundant pairs from $G_{i,j}$ using the $\{P_{i_k,j_k} : k = 1, 2, \ldots, r_{ij}\}$. (The redundant pairs are generated because of ambiguity of cross-angles and splitting of a facet in extraction processing from real 3-D range data.) This elimination is carried out by detecting the pair with the maximum P_{i_k,j_k} and eliminating other pairs among the set of pairs with the same facet number of the model. The set $G_{i,j}$ will become the so-called "kernel"-$G_{i,j}$, in short $KG_{i,j}$, after elimination of redundant pairs.

(5) Deleting the representative pair (i, j) and returning to point (1) if the value $Q_{i,j}/M_m$ (will be defined later) is smaller than a given threshold value, otherwise the representative pair (i, j) is regarded as a correct one; that is, the scene including the facet j and the model including the facet i are recognized as the same object category. The value of $Q_{i,j}/M_m$ is another measure for a pair (i, j) to represent a likelihood of matching and is defined by using:

$$Q_{i,j} \triangleq \left| \bigcap_{\ell \in K^3} L_{i_\ell, j_\ell} \right|$$

$$L_{i_\ell, j_\ell} \triangleq \{\underline{i}: (\underline{i}, \underline{j}) \in KG_{i_\ell, j_\ell}\}, \quad (\ell = 1, 2, \ldots, g_{i,j})$$

where $\bigcap_{\ell \in K^3} L_{i_\ell, j_\ell}$ means the intersection of the largest three sets of facets of a model included in KG_{i_ℓ, j_ℓ}, and M_m indicates the total number of facets included in the model object numbered n. Let us represent the largest three sets among $KG_{i,j}$ by the set of (i_ℓ, j_ℓ)'s, and define the set by the symbol K^3.

A set L_{i_ℓ, j_ℓ} means the facets of a model each of which matches with a facet of a scene without duplication. A set $\bigcap_{\ell \in K^3} L_{i_\ell, j_\ell}$ means facets of a model all of which satisfy the constraint by three facets with respect to relative locations and cross- angles each other. Therefore, the total facets $\bigcap_{\ell \in K^3} L_{i_\ell, j_\ell}$ divided by M_m can be regarded as a likelihood of matching by the pair (i, j).

[end of verification algorithm]

According to the ambiguity reduction using some A_i's, three A_i's are not enough to reconstruct an original object. However, in this verification algorithm three A_i's are used to determine the possible facets constituting an object. Despite this condition which cannot completely eliminate the ambiguity, the experimental results lead to the conclusion that the verification algorithm works effectively for recognition problems of real 3-D objects.

8. EXPERIMENTAL RESULTS OF 3-D OBJECTS RECOGNITION

In order to evaluate the usefulness of the Cross-Angle Transform in real recognition problems of 3-D objects, we will show some recognition results. The problem is to recognize each blob in a scene as one of the model objects when each of them is represented by a set of A_i's with $\alpha\beta$-histograms. The model representations are made by showing the 3-D picture of Fig. 13 to the machine of which separated blob corresponds to a different model. The blobs in the Fig. 14 are the objects to be recognized. Therefore we have to say that a description of a 3-D object is produced by a set of facets obtained by a single view data of a model or a scene. The

Figure 13. The three-dimensional data for learning the model descriptions by using the Cross-Angle Transform.

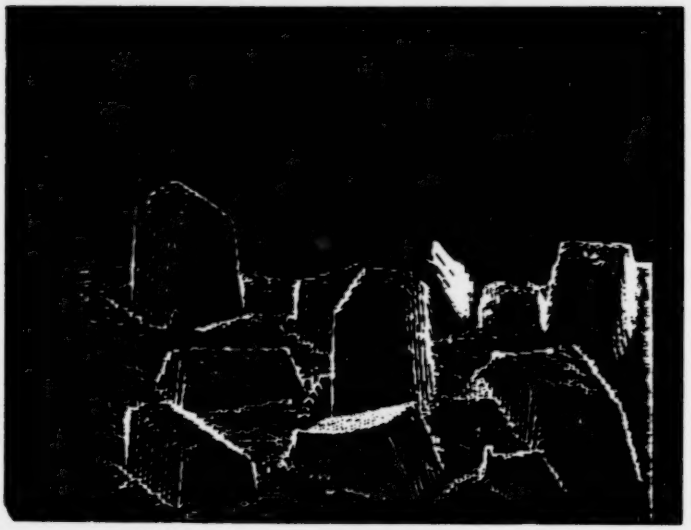


Figure 14. The three-dimensional data to be recognized.

segmentation algorithm described in Chapter 1 is applied to the two 3-D pictures and results in the facet representations of the blobs as shown in Figs. 15 and 16. The blob numbers and the facet numbers in each blob are shown in Figs. 17 and 18.

In this experiment, cross-angles α, β are quantized into 8, 7 levels as shown in Fig. 19, respectively.

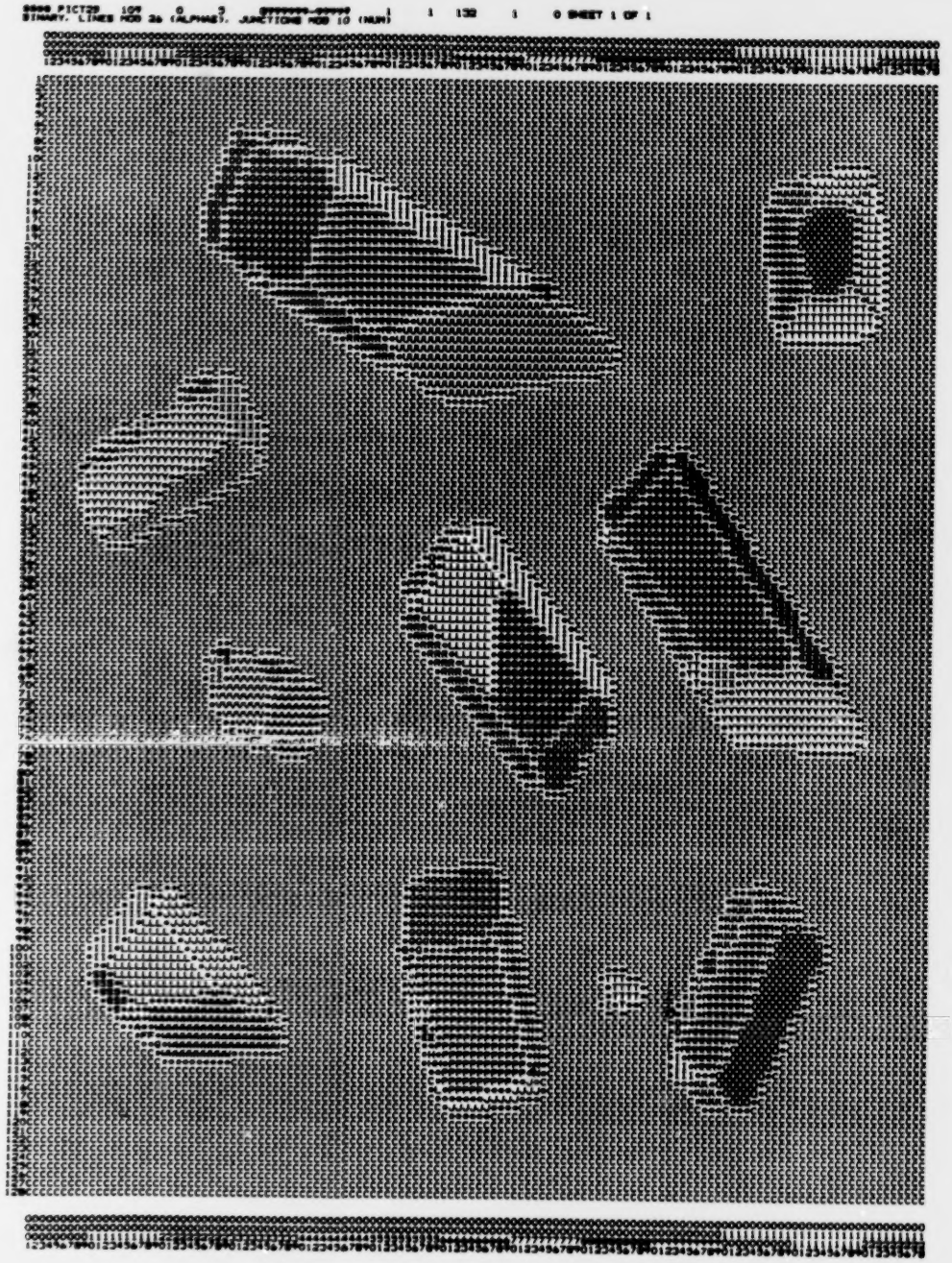


Figure 15. The facet segmentation of the three-dimensional data of Fig. 13.

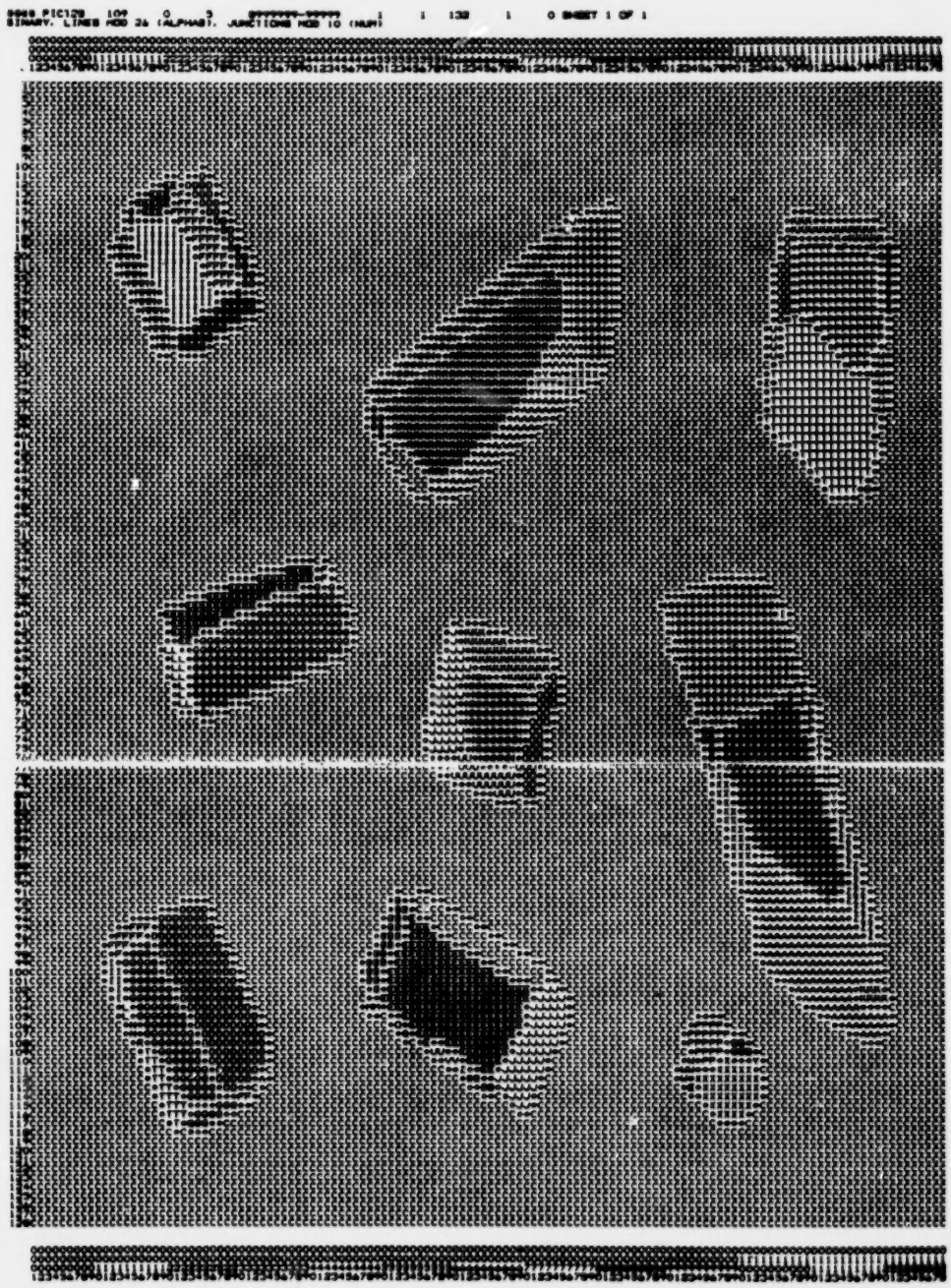


Figure 16. The facet segmentation of the three-dimensional data of Fig. 14.

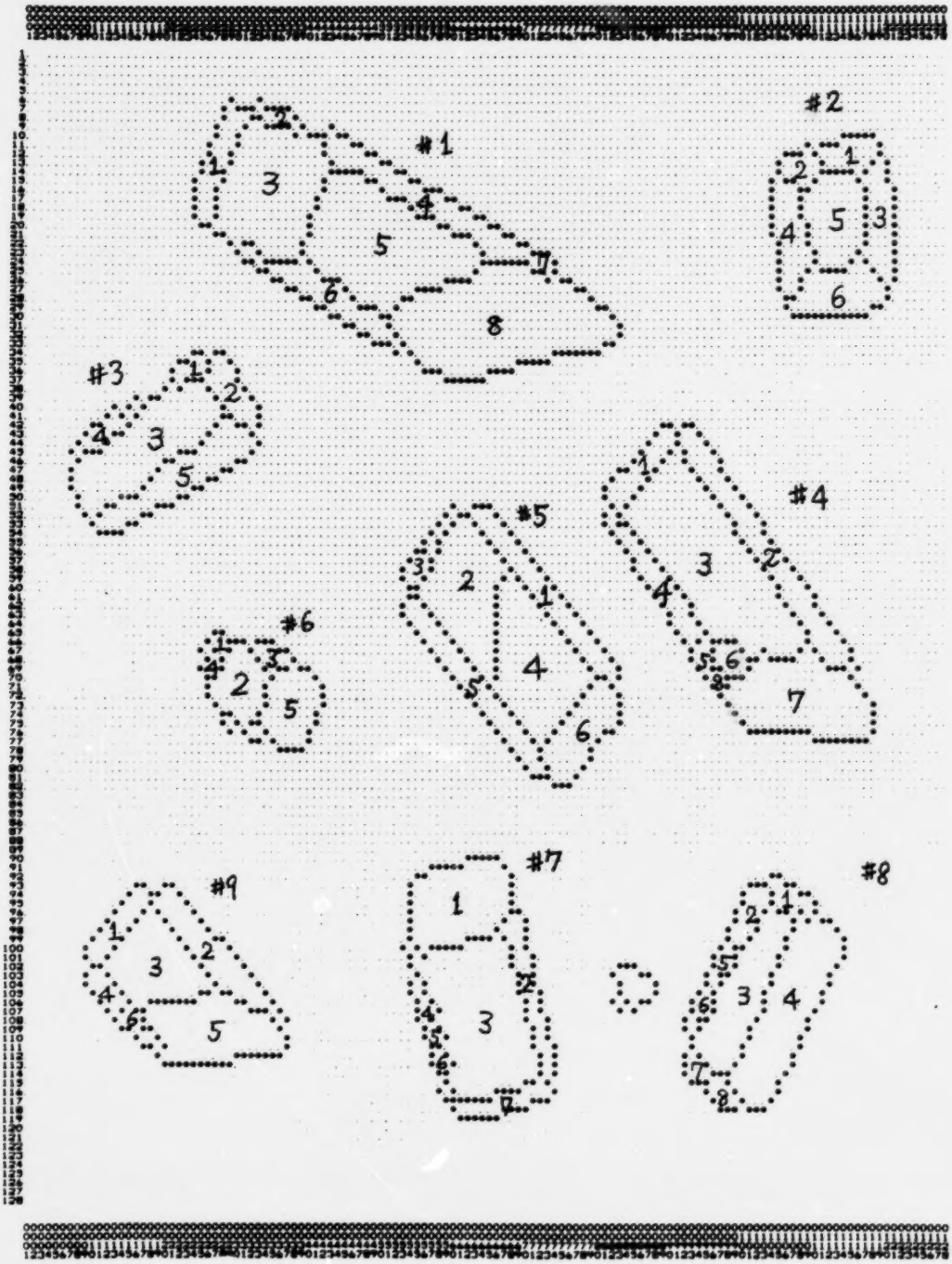


Figure 17. The sub-scene numbers (#) and the facet numbers of each sub-scene of Fig. 15.

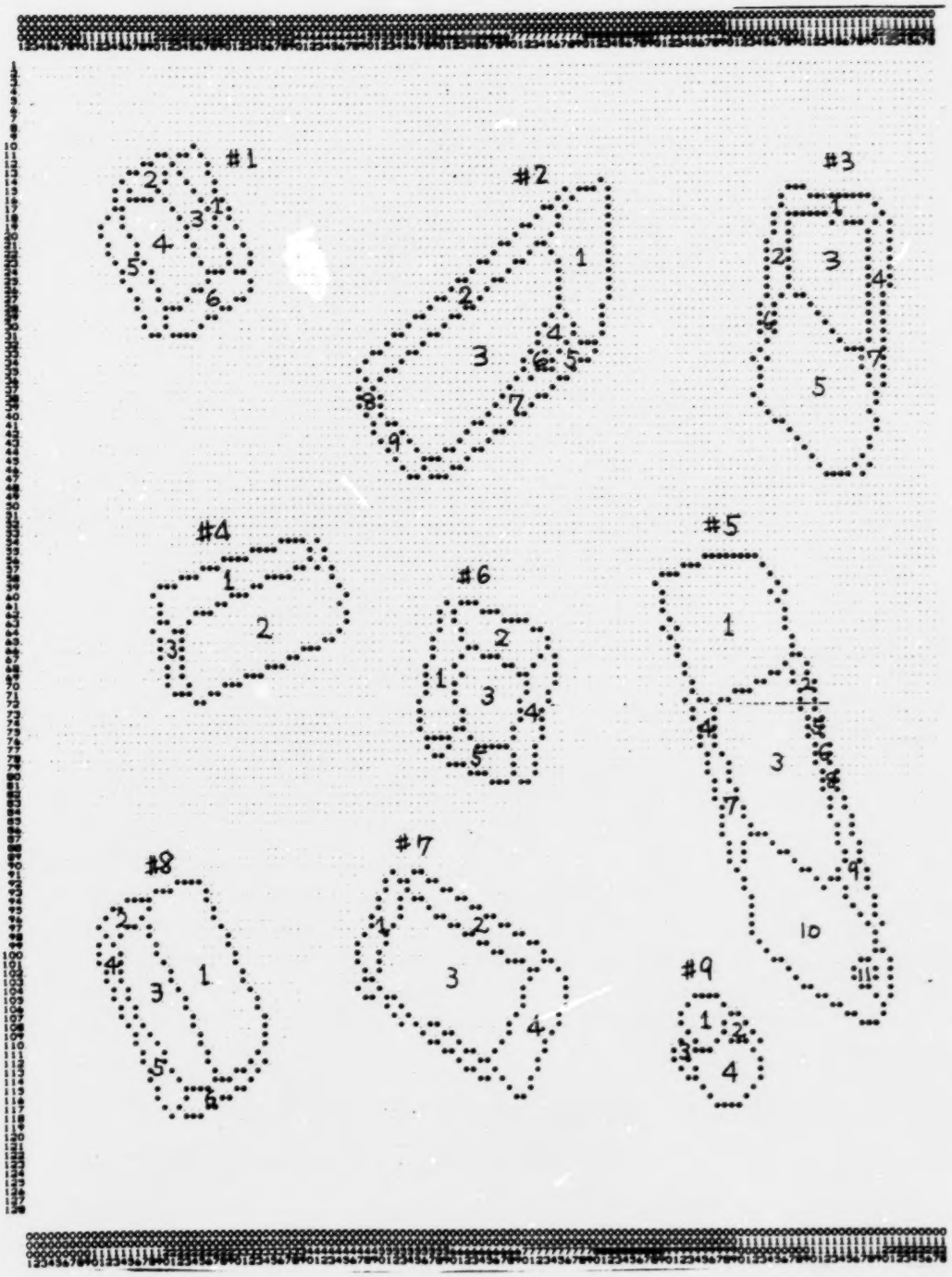


Figure 18. The sub-scene numbers (#) and the facet numbers of each sub-scene of Fig. 16.

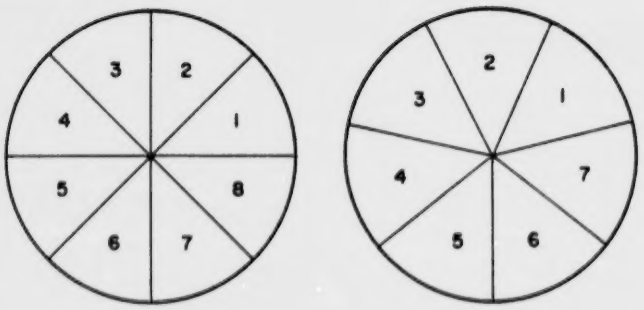


Figure 19. The quantizations for (a) the cross-angle α and (b) and the cross-angle β.

Table 2 shows the distance matrix between the three largest facets of each model and the facet numbers of each blob which gives the minimum distance value for all the blobs in the 3-D picture of Fig. 18.

In Table 3 we can see that (1) one pair (called the "father" pair) composed of a facet in a model and a facet in a blob of the scene is selected for each model by detecting the minimum distance among 27 (= 3 × 9) pairs in Table 2, (2) the set $G_{i,j}$ produced by the "father" pair (i, j), where i and j indicate representative facets of the model and the blob of scene, (3) the i_k, i_k values of the elment pairs of $G_{i,j}$, where "ALPHA BETA" indicates the pairs of the cross angles α_{i_k}, β_{i_k}.

Table 3 shows the stage of the determination of the first representative facet pair for each model before the application of the verification algorithm (see Fig. 1). This stage of Table 3 does not imply the final recognition stage; however, we can see that the models numbered 1, 2, 3, 6, 7, 8 are recognized correctly and the models 4, 5, 9 are not recognized correctly. In the following, let us see how the verifiation algorithm works in some examples.

8.1 An Example of the Parameters Used for Verification

Table 4 shows an example of "children" G_{i_k,j_k}'s produced by $G_{i,j}$, namely, a set of "brother of father"s, for i = 8 (facet number) in the model number 1 and j = 10 (facet number) in the blob number 5 of the scene. This pair is obtained because $D_{i=8,j=10}$ gives the minimum value. The "father" pair (i, j) = (8, 10) is supported by the 12 ($=r_{ij}$) "brother of father"s; (1, 9), (2, 2), (2, 5), (2, 6), (2, 8), (3, 1), (4, 9), (5, 3), (6, 4), (6, 7), (6, 11), (8, 10). Moreover, we can obtain the set of facets {1, 2, 3, 4, 5, 6, 8} whose elements are different from each other

Table 2. Distance matrix between the three largest facets of each model and the facets of all the blobs of input

OBJECT	BLOB=	1 (FCT, DIS)	2 (FCT, DIS)	3 (FCT, DIS)	4 (FCT, DIS)	5 (FCT, DIS)	6 (FCT, DIS)	7 (FCT, DIS)	8 (FCT, DIS)	9 (FCT, DIS)
MDL= 1	FACET= 8	(3,1378)	(3,1135)	(5, 959)	(2, 939)	(10, 440)	(3,1311)	(3,1619)	(1,1050)	(4,1240)
MDL= 1	FACET= 3	(6,1235)	(5,1143)	(5,1674)	(2,1338)	(7, 832)	(2, 885)	(3,1673)	(6,1155)	(1,1136)
MDL= 1	FACET= 5	(1,1952)	(3, 707)	(3, 601)	(2,1270)	(3, 524)	(3,1465)	(1,1306)	(3, 729)	(4,1210)
MDL= 2	FACET= 6	(3,1253)	(5,1055)	(7,1182)	(2, 840)	(9,1208)	(3, 537)	(3,1308)	(1,1024)	(4, 760)
MDL= 2	FACET= 3	(2, 946)	(2,1117)	(7, 906)	(1, 825)	(9,1055)	(4, 471)	(4,1015)	(2,1094)	(4, 676)
MDL= 2	FACET= 4	(2, 434)	(2, 799)	(7,1070)	(1, 683)	(1,1137)	(1, 209)	(4, 939)	(6, 642)	(4, 676)
MDL= 3	FACET= 5	(1, 973)	(3,1139)	(5, 948)	(2, 383)	(10,1110)	(3, 609)	(3,1108)	(4, 733)	(4, 412)
MDL= 3	FACET= 3	(6, 755)	(7, 990)	(4,1056)	(2, 636)	(1, 993)	(4, 442)	(3, 999)	(6, 663)	(4, 328)
MDL= 3	FACET= 2	(3,1401)	(8,1170)	(1,1044)	(3, 131)	(6,1614)	(3, 899)	(3,1302)	(2, 835)	(4, 416)
MDL= 4	FACET= 7	(1,1109)	(1, 819)	(5, 702)	(2, 671)	(10,1009)	(4,1631)	(3,1391)	(1, 670)	(4, 990)
MDL= 4	FACET= 3	(4,1741)	(3, 485)	(3, 397)	(2, 962)	(3, 602)	(3,1419)	(3,1351)	(3, 465)	(4, 942)
MDL= 4	FACET= 2	(6, 876)	(2,1024)	(4, 538)	(2, 671)	(1,1201)	(4, 587)	(3,1506)	(6, 970)	(4, 888)
MDL= 5	FACET= 1	(2, 868)	(2, 758)	(4, 866)	(2, 806)	(1, 936)	(5, 188)	(3,1453)	(2, 759)	(4, 721)
MDL= 5	FACET= 5	(2,1028)	(2, 996)	(2,1039)	(2, 977)	(1, 681)	(4, 342)	(3,1455)	(1, 895)	(4, 721)
MDL= 5	FACET= 6	(6, 811)	(7,1009)	(4,1326)	(2,1021)	(1, 880)	(2, 450)	(3,1499)	(6, 720)	(4, 721)
MDL= 6	FACET= 5	(6, 660)	(2,1163)	(7, 783)	(1, 395)	(9,1185)	(3, 635)	(4, 661)	(6, 625)	(4, 80)
MDL= 6	FACET= 2	(6, 867)	(2,1183)	(4, 917)	(2, 528)	(1,1317)	(2, 641)	(3,1070)	(6, 571)	(4, 88)
MDL= 6	FACET= 3	(6, 833)	(2,1147)	(4, 898)	(2, 662)	(1,1175)	(2, 607)	(3,1140)	(6, 505)	(1, 120)
MDL= 7	FACET= 3	(6, 900)	(3,1147)	(5, 994)	(1, 355)	(3,1687)	(4,1315)	(3, 204)	(1, 709)	(4, 644)
MDL= 7	FACET= 1	(6, 422)	(2,1408)	(7, 969)	(1, 355)	(1,1499)	(4, 906)	(4, 204)	(1, 928)	(1, 590)
MDL= 7	FACET= 2	(3,1149)	(8,1455)	(7,1192)	(3, 646)	(6,1113)	(3, 621)	(3,1462)	(2, 961)	(4, 656)
MDL= 8	FACET= 4	(6,1338)	(3, 600)	(5, 429)	(2, 511)	(10, 802)	(3, 916)	(3,1047)	(1, 100)	(4, 491)
MDL= 8	FACET= 3	(4,1355)	(3, 658)	(3, 215)	(2, 783)	(3, 637)	(4,1418)	(3,1152)	(3, 68)	(4, 509)
MDL= 8	FACET= 1	(2, 713)	(2,1033)	(4, 940)	(3, 749)	(1,1304)	(2, 348)	(3,1453)	(6, 32)	(1, 455)
MDL= 9	FACET= 5	(1, 916)	(1, 926)	(5, 337)	(2, 262)	(10, 976)	(4,1206)	(3, 839)	(1, 375)	(4, 565)
MDL= 9	FACET= 2	(6, 969)	(3,1540)	(7, 658)	(1, 262)	(9,1060)	(3, 940)	(2, 366)	(5, 807)	(4, 547)
MDL= 9	FACET= 3	(4,1185)	(3, 770)	(3, 467)	(2, 815)	(3, 889)	(4,1450)	(3,1020)	(3, 320)	(4, 535)

Table 3. The pairs of the model facet and the scene facet which give the minimum distances among the blobs in Table 2. The symbol X indicates the correct (incorrect) recognition based only on the distances.

ORDER OF THE CANDIDATE= 1

	"FATHER" PAIR	"BROTHER OF FATHER" PAIR														
MODEL BLOB = 1 FACET= 8 (DIS= 440)	1	2	2	2	2	3	4	5	6	6	6	8	0	0	0	0
SCENE BLOB = 5 FACET= 10 (DIS= 440)	9	2	5	6	8	1	9	3	4	7	11	10	0	0	0	0
ALPHA BETA=55	65	65	65	65	45	55	35	24	24	24	11	00	00	00	00	
MODEL BLOB = 2 FACET= 4 (DIS= 209)	3	4	5	5	5	6	6	6	0	0	0	0	0	0	0	0
SCENE BLOB = 6 FACET= 1 (DIS= 209)	4	1	2	3	5	2	3	5	0	0	0	0	0	0	0	0
ALPHA BETA=77	11	46	46	46	46	46	46	00	00	00	00	00	00	00	00	
MODEL BLOB = 3 FACET= 2 (DIS= 131)	2	3	4	5	0	0	0	0	0	0	0	0	0	0	0	0
SCENE BLOB = 4 FACET= 3 (DIS= 131)	3	2	2	1	0	0	0	0	0	0	0	0	0	0	0	0
ALPHA BETA=11	47	47	57	00	00	00	00	00	00	00	00	00	00	00	00	
MODEL BLOB = 4 FACET= 3 (DIS= 397)	1	1	1	1	2	2	2	2	3	4	4	4	4	5	7	8
SCENE BLOB = 3 FACET= 3 (DIS= 397)	1	2	4	7	1	2	4	7	3	1	2	4	7	6	5	6
ALPHA BETA=44	44	44	44	44	44	44	44	11	44	44	44	44	45	35	45	
MODEL BLOB = 5 FACET= 1 (DIS= 188)	1	2	2	4	4	5	6	0	0	0	0	0	0	0	0	0
SCENE BLOB = 6 FACET= 5 (DIS= 188)	5	3	4	3	4	2	1	0	0	0	0	0	0	0	0	0
ALPHA BETA=11	46	46	46	46	77	45	00	00	00	00	00	00	00	00	00	
MODEL BLOB = 6 FACET= 5 (DIS= 80)	2	3	5	0	0	0	0	0	0	0	0	0	0	0	0	0
SCENE BLOB = 9 FACET= 4 (DIS= 80)	1	3	4	0	0	0	0	0	0	0	0	0	0	0	0	0
ALPHA BETA=46	45	11	00	00	00	00	00	00	00	00	00	00	00	00	00	
MODEL BLOB = 7 FACET= 3 (DIS= 204)	1	3	4	5	6	7	0	0	0	0	0	0	0	0	0	0
SCENE BLOB = 7 FACET= 3 (DIS= 204)	4	3	2	2	2	1	0	0	0	0	0	0	0	0	0	0
ALPHA BETA=56	11	34	34	34	24	00	00	00	00	00	00	00	00	00	00	
MODEL BLOB = 8 FACET= 1 (DIS= 32)	1	2	2	3	3	4	4	5	5	6	6	7	8	0	0	0
SCENE BLOB = 8 FACET= 6 (DIS= 32)	6	1	5	3	4	1	5	1	5	3	4	2	2	0	0	0
ALPHA BETA=11	46	46	47	47	46	46	46	46	47	47	77	77	00	00	00	
MODEL BLOB = 9 FACET= 5 (DIS= 262)	2	5	0	0	0	0	0	0	0	0	0	0	0	0	0	0
SCENE BLOB = 4 FACET= 2 (DIS= 262)	1	2	0	0	0	0	0	0	0	0	0	0	0	0	0	0
ALPHA BETA=55	11	00	00	00	00	00	00	00	00	00	00	00	00	00	00	

Table 4. An example of the configuration of the "children" facet pairs in order to verify the correspondence of Table 2 for recognizing model number 1. This table shows the "children" pairs for the eighth facet of model 1 and the tenth facet of blob 5 (i, j) = 8, 10).

MODEL NUMBER= 8

BROTHER OF FATHER PAIR	(MODEL)= 1 (SCENE)= 6	CHILDREN PAIR =	1 2 2 3 3 4 4 5 5 6 6 7 8 0 0 0 0 0 0 0 0 0 6 1 5 3 4 1 5 1 5 3 4 2 2 0 0 0 0 0 0 0 0 0
BROTHER OF FATHER PAIR	(MODEL)= 2 (SCENE)= 1	CHILDREN PAIR =	2 3 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 1 3 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
BROTHER OF FATHER PAIR	(MODEL)= 2 (SCENE)= 5	CHILDREN PAIR =	1 1 1 2 4 5 6 7 7 7 0 0 0 0 0 0 0 0 0 0 0 2 3 6 5 1 4 4 2 3 6 0 0 0 0 0 0 0 0 0 0 0
BROTHER OF FATHER PAIR	(MODEL)= 3 (SCENE)= 3	CHILDREN PAIR =	1 1 1 1 3 4 5 5 5 5 6 6 6 6 7 7 7 7 8 8 8 8 2 4 5 6 3 1 2 4 5 6 2 4 5 6 2 4 5 6 2 4 5 6
BROTHER OF FATHER PAIR	(MODEL)= 3 (SCENE)= 4	CHILDREN PAIR =	1 3 5 6 7 8 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 6 4 6 6 6 6 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
BROTHER OF FATHER PAIR	(MODEL)= 4 (SCENE)= 1	CHILDREN PAIR =	2 2 3 4 5 5 6 6 8 8 0 0 0 0 0 0 0 0 0 0 0 4 5 3 1 4 5 4 5 2 6 0 0 0 0 0 0 0 0 0 0 0
BROTHER OF FATHER PAIR	(MODEL)= 4 (SCENE)= 5	CHILDREN PAIR =	1 1 1 4 7 7 7 0 0 0 0 0 0 0 0 0 0 0 0 0 0 2 3 6 5 2 3 6 0 0 0 0 0 0 0 0 0 0 0 0 0 0
BROTHER OF FATHER PAIR	(MODEL)= 5 (SCENE)= 1	CHILDREN PAIR =	5 7 7 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 1 2 6 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
BROTHER OF FATHER PAIR	(MODEL)= 5 (SCENE)= 5	CHILDREN PAIR =	1 1 1 2 4 5 6 0 0 0 0 0 0 0 0 0 0 0 0 0 0 2 3 6 4 1 5 4 0 0 0 0 0 0 0 0 0 0 0 0 0 0
BROTHER OF FATHER PAIR	(MODEL)= 6 (SCENE)= 3	CHILDREN PAIR =	6 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 3 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
BROTHER OF FATHER PAIR	(MODEL)= 6 (SCENE)= 4	CHILDREN PAIR =	1 2 3 4 5 6 7 0 0 0 0 0 0 0 0 0 0 0 0 0 0 3 5 2 1 5 4 3 0 0 0 0 0 0 0 0 0 0 0 0 0 0
BROTHER OF FATHER PAIR	(MODEL)= 7 (SCENE)= 2	CHILDREN PAIR =	1 2 2 3 3 4 5 6 7 0 0 0 0 0 0 0 0 0 0 0 0 6 3 5 3 5 1 1 1 2 0 0 0 0 0 0 0 0 0 0 0 0
BROTHER OF FATHER PAIR	(MODEL)= 8 (SCENE)= 2	CHILDREN PAIR =	1 3 3 4 4 8 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 6 3 5 3 5 2 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0

and also are included in a model, so that $g_{8,10} = 7$ is obtained. At this stage, we can see that there are two sets which indicate one-to-many correspondence with respect to correspondence between facets of a model and a scene, that is, the sets: $\{(2, 2), (2, 5), (2, 6), (2, 8)\}$, $\{(6, 4), (6, 7), (6, 11)\}$.

In general, the pair set constituting a one-to-many correspondence is produced in this system by the effects of those: (1) ambiguity of the Cross-Angle Transform and (2) splitting of a facet in extraction processing of a facet from the real 3-D range data. We can see from Figs. 17 and 18 that the facets 2, 5, 6, 8 of the scene are regarded as the split ones of a facet which should correspond to the facet 2 of the model 1, while the facets 4, 7 of a scene correspond to the facet 6 of the model 1. On the other hand, we can see that the facet 11 of the scene appeared in the pair (6, 11) because of the ambiguity of the Cross-Angle Transform.

Let us assume that the one-to-one correspondence is correct with respect to the correspondences between facets of a model and a scene. Then we will consider the determination of a correct pair in a set of pairs indicating one-to-many correspondence by using the matching likelihood measure P_{i_k,j_k}'s. Let us count P_{i_k,j_k} $(k = 1, 2, \ldots, 12)$ in order to determine which pair is redundant in the set $\{(2, 2), (2, 5), (2, 6), (2, 8)\}$. We can obtain $P_{2,2} = 4$ for (2, 2), $P_{2,5} = 3$ for (2, 5), $P_{2,6} = 4$ for (2, 6), and $P_{2,8} = 4$ for (2, 8). The values P_{i_k,j_k}'s for (2, 2), (2, 6), (2, 8) are equal to each other and indicate the maximum, so that we will select (2, 2) and eliminate others. We can apply the same elimination process to the set $\{(6, 4), (6, 7), (6, 11)\}$ and eliminate (6, 4), (6, 11). In this case $P_{6,7}$ gives the maximum value uniquely. Thus we can obtain $KG_{8,10}$ as follows:

$$KG_{8,10} = \{(1,9), (2,2), (3,1), (4,9), (5,3), (6,7), (8,10)\}$$

where

$$P_{i_k,j_k}: \quad k = 1, 2, 6, 7, 8, 10, 12 = 4, 4, 5, 4, 6, 7, 7$$

Now let us show the values of the parameters, (i_ℓ, j_ℓ), L_{i_ℓ,j_ℓ} $(\ell = 1, 2, \ldots, g_{i,j})$, $\left|L_{i_\ell,j_\ell}\right|$, K^3, $\left|L_{i_\ell,j_\ell}\right|$, $Q_{i,j}$ and M_m, by the following.

$$(i_1,j_1) = (1,9), \; L_{i_1,j_1} = \{1,4,6,8\}, \; \left|L_{i_1,j_1}\right| = 4$$

$$(i_2,j_2) = (2,2), \; L_{i_2,j_2} = \{2,5,6,8\}, \; \left|L_{i_2,j_2}\right| = 4$$

$$(i_3,j_3) = (3,1), \; L_{i_3,j_3} = \{3,4,6,7,8\}, \; \left|L_{i_3,j_3}\right| = 5$$

$(i_4,j_4) = (4,9)$, $L_{i_4,j_4} = \{1,3,4,7\}$, $|L_{i_4,j_4}| = 4$

$(i_5,j_5) = (5,3)$, $L_{i_5,j_5} = \{1,3,4,5,7,8\}$, $|L_{i_5,j_5}| = 6$

$(i_6,j_6) = (6,7)$, $L_{i_6,j_6} = \{1, 2,3,4,5,6,8\}$, $|L_{i_6,j_6}| = 7$

$(i_7,j_7) = (8,10)$, $L_{i_7,j_7} = \{1, 2,3,4,5,6,8\}$, $|L_{i_7,j_7}| = 7$

$K^3 = \{(i_5,j_5), (i_6,j_6), (i_7,j_7)\} = \{(5,3), (6,7), (8,10)\}$

$\bigcap_{\ell \in R^3} L_{i_\ell,j_\ell} = \{1,3,4,5,8\}$

$Q_{i,j} = 5$

$M_1 = 8$

A value $Q_{i,j}/M_m$ with respect to a representative pair (i, j) is used for one trial verification of whether or not the model including the ith facet and the blob in a scene including the jth facet belong to the same category. In other words, the verification algorithm works to verify the selection of a representative pair by using $D_{i,j}$ value, since the distance value $D_{i,j}$ cannot always give the exact matching level.

Thus repeated applications of the verification algorithm sometimes lead to a representative pair which does not give the minimum value among $D_{i,j}$'s as a correct one.

8.2 An Example of Repeated Verification Process

Let us see repeated applications of verification algorithm in the following example as shown in Table 5.

We can select nine pairs based on 27 $D_{i,j}$'s with respect to the model 9 such that each of these pairs belongs to a different blob and also gives the minimum value among three $D_{i,j}$'s related to the same blob. Moreover, we select the smallest three among the nine $D_{i,j}$'s of the pairs. The three representative pairs are the following:

((i,j) = (5,2): model = 9, blob = 4, $D_{5,2} = 262$)

((i,j) = (3,3): model = 9, blob = 8, $D_{3,3} = 320$)

((i,j) = (5,5): model = 9, blob = 3, $D_{5,5} = 337$)

Table 5. The generation of the "children" facets for three candidate representative pairs for recognizing the model 9. Verification algorithm is applied to each representative pair.

BROTHER OF FATHER FACET PAIR (F-MODEL,F-SCENE)	CHILDREN FACET PAIR (F-MODEL,F-SCENE)
(I) FIRST CANDIDATE BLOB = 4, $D_{i,j}$ = 262	
(2,1)	(2,1),(5,2)
(5,2)	(2,1),(5,2)
(II) SECOND CANDIDATE BLOB = 8, $D_{i,j}$ = 320	
(3,3)	(3,3),(4,2),(4,4),(4,5),(4,6) (5,1),(6,2),(6,4),(6,5),(6,6)
(4,2)	(1,1),(3,1),(4,2)
(4,4)	(1, 2),(2,1),(3,2),(4,4)
(4,5)	(4,5)
(4,6)	(1,1),(1,5),(3,1),(3,5),(4,6)
(5,1)	(1,4),(1,5),(3,3),(5,1)
(6,2)	(3,1),(6,2)
(6,4)	(2,1),(3,2),(6,4)
(6,5)	(6,5)
(6,6)	(3,1),(3,5),(6,6)
(III) THIRD CANDIDATE BLOB = 3, $D_{i,j}$ = 337	
(1,1)	(1,1),(2,2),(2,7),(4,4),(5,5),(6,6)
(2,4)	(2,4)
(3,3)	(3,3),(4,1),(4,2),(4,4),(4,7), (5,5),(6,1),(6,2),(6,4),(6,7)
(4,2)	(1,1),(3,1),(4,2),(5,5)
(4,6)	(1,3),(3,3),(4,6)
(5,5)	(1,1),(2,4),(3,3),(4,2),(4,6),(5,5)

Verification is carried out in the order (5,2), (3,3), and (5,5) because $D_{5,2} < D_{3,3} < D_{5,5}$.

For the representative pair (i, j) = (5, 2), the first verification process gives the following data,

$$(i_1,j_1) = (2,1), \quad L_{i_1,j_1} = \{2,5\}, \quad \left|L_{i_1,j_1}\right| = 2$$

$$(j_2,j_2) = (5,2), \quad L_{i_2,j_2} = \{2,5\}, \quad \left|L_{i_2,j_2}\right| = 2$$

$$K^3 = \{(2,1),(5,2)\}$$

$$\bigcap_{\ell \in R^3} L_{i_\ell, j_\ell} = \{2,5\}$$

$$Q_{5,2} = 2$$

$$Q_{5,2}/M_9 = 2/6 = 0.33$$

For the representative pair (i, j) = (3,3), the second verification process gives the following data

$$(i_1,j_1) = (3,3), \quad L_{i_1,j_1} = \{3,4,5,6\}, \quad \left|L_{i_1,j_1}\right| = 4$$

$$(i_2,j_2) = (4,4), \quad L_{i_2,j_2} = \{1, 2,3,4\}, \quad \left|L_{i_2,j_2}\right| = 4$$

$$(i_3,j_3) = (5,1), \quad L_{i_3,j_3} = \{1,3,5\}, \quad \left|L_{i_3,j_3}\right| = 3$$

$$(i_4,j_4) = (6,4), \quad L_{i_4,j_4} = \{2,3,6\}, \quad \left|L_{i_4,j_4}\right| = 3$$

$$K^3 = \{(3,3),(4,3),(5,1)\}$$

$$\bigcap_{\ell \in R^3} L_{i_1,j_1} = \{3\}$$

$$Q_{5,2} = 1$$

$$Q_{3,3}/M_9 = 1/6 = 0.166$$

For the representative pair (i, j) = (5, 5), the third verification process gives the following data:

$$(i_1,j_1) = (1,1), \quad L_{i_1,j_1} = \{1,2,4,5,6\}, \quad \left|L_{i_1,j_1}\right| = 5$$

$$(i_2,j_2) = (2,4), \quad L_{i_2,j_2} = \{2\}, \quad \left|L_{i_2,j_2}\right| = 1$$

$$(i_3,j_3) = (3,3), \quad L_{i_3,j_3} = \{3,4,5,6\}, \quad \left|L_{i_3,j_3}\right| = 4$$

$$(i_4,j_4) = (4,2), \quad L_{i_4,j_4} = \{1,3,4,5\}, \quad \left|L_{i_4,j_4}\right| = 4$$

$$(i_5,j_5) = (5,5), \quad L_{i_5,j_5} = \{1,2,3,4,5\}, \quad \left|L_{i_5,j_5}\right| = 5$$

$$K^3 = \{(1,1), (3,3), (5,5)\} \text{ or } \{(1,1), (4,2), (5,5)\}$$

$$\bigcap_{\ell \in R^3} L_{i_1,j_1} = \{1,4,5\}$$

$$Q_{5,2} = 3$$

$$Q_{5,5}/M_9 = 3/6 = 0.500$$

Here we need two parameters, that is, the number of allowed verification times and the threshold value for $Q_{i,j}/M_m$. If we set a maximum of three verification times and the threshold value 0.45, the verification will stop after three verifications and lead to the answer that the pair (5, 5) is correct; that is, the blob numbered 3 is recognized as the model numbered 9.

8.3 Analysis of Verification Process

Here, we can see that this decision is certainly correct and the verification algorithm is working effectively to check the primary recognition by the distance between histograms.

Let us show the results in Table 6 obtained by the application of the verification algorithm to the nine blobs in Fig. 18 in order to identify the model objects in Fig. 17. For the nine blobs in Fig. 18, the models numbered 1, 2, 6, 7, 8, 9 are correctly recognized, the blob numbered 5 is incorrectly recognized, and there are no decisions for the models numbered 3 and 4. We will examine the details concerning the models numbered 3, 4, and 5.

8.3.1 Analysis of the Verification Applied to Model 3

The first example is concerned with model 3. In this example, the result of the verification for recognizing model 3 indicates the incorrect result that there is no blob in the Fig. 18.

The first selection of the representative pair yields the facet pair (3, 2) composed of a facet 2 of the model number 3 and a facet 3 of the scene number 4. This is a correct pair, but the primary decision is not verified because $Q_{3,2} = 0$. The representative pair (3, 2) produces a set of "brother of father"s: $\{$(model facet, scene facet)$\}$ = $\{$(2, 3), (3, 2), (4, 2), (5, 1)$\}$. We can see that the two facets 3 and 4 of model 3 correspond to the facet 2 of scene 4. This is a correct assertion, because it is a fact that the facet extraction error has been caused by dividing one facet into two facets when making model description of the model 3.

Table 6. The data used for repeated verification

	M_i	Verification Order	Object Number	D_{ij}	G_{ij}	G_{ij}/M_i
model=1	8	1	5	440	5	0.63
		2	3	601	4	0.50
		3	2	707	4	0.50
model=2	6	1	6	209	4	0.67
		2	1	434	4	0.67
		3	8	642	3	0.50
model=3	5	1	4	131	0	0.0
		2	9	328	0	0.0
		3	6	442	0	0.0
model=4	8	1	3	397	5	0.63
		2	8	465	4	0.50
		3	2	485	8	1.00
model=5	6	1	6	188	2	0.34
		2	5	681	4	0.67
		3	8	720	1	0.16
model=6	5	1	9	80	2	0.40
		2	4	395	0	0.0
		3	8	505	0	0.0
model=7	7	1	7	204	5	0.71
		2	4	355	1	0.14
		3	1	422	1	0.14
model=8	8	1	8	32	7	0.85
		2	3	215	6	0.67
		3	6	348	2	0.25
model=9	6	1	4	262	2	0.34
		2	8	320	1	0.16
		3	3	337	3	0.50

However, we have $Q_{3,2} = 0$, that is, $\bigcap_{\ell \in R^3} L_{i_\ell, j_\ell} = \{0\}$, while other "brother of father"s, (2, 3), (5, 1), indicate the correct correspondences. The pair (5, 1) (a "brother of father") has reasonable children (2, 3), (5, 1), so that the verification has failed because of the failure of children production by the pair (3, 2) or (4, 2). In fact, the cross-angles for the pair (3, 2) are,

model facet = 3 --- $(\alpha_{3,2}, \beta_{3,2}) = (4,5)$, $(\alpha_{3,5}, \beta_{3,5}) = (4,6)$

scene facet = 2 --- $(\alpha_{2,3}, \beta_{2,3}) = (4,4)$, $(\alpha_{2,1}, \beta_{2,1}) = (5,5)$

where cross-angles are represented by numbers of quantized directions as shown in Fig. 19.

Thus we can conclude that the facet 4 in the model 3 has prevented the pair (3, 2) from producing reasonable children (2, 3), (5, 1). The pair (3, 2) would produce children (2, 3) and (5, 1) if $(\alpha_{3,2}, \beta_{3,2}) = (\alpha_{2,3}, \beta_{2,3})$ and $(\alpha_{3,5}, \beta_{3,5}) = (\alpha_{2,1}, \beta_{2,1})$ could hold. The situation concerning the pair (4, 2) is the same. This means that three features (the normal vector, the location, and the area) of a facet are sometimes disturbed critically by facet splitting.

Other representative pairs for further verification are, of course, under more severe conditions because of incorrect primary selections.

This example gives us the important fact that the splitting of a facet sometimes causes unreasonable reduction of the likelihood of matching, especially when the total number of facets included in a model is small.

8.3.2 Analysis of the Verification Applied to the Model 4

In this example, the verification for recognizing the model number 4 leads to the incorrect result that there are no blobs which correspond to the model number 4 in Fig. 18.

This example has characteristics such that the correct result will appear at the stage of the third verification if the verification can be repeated. The third verification gives the value of Q_{ij}/M_i as 1.00 for the scene object numbered 2; this is regarded as a correct result. However, the preceding two verifications give the Q_{ij}/M_i values which exceed the threshold value 0.4. This example suggests that the verification will fail if there is no procedure in the verification algorithm to allow the succeeding verifications to also give the final result even if the preceding verification gives a Q_{ij}/M_i value exceeding the threshold value.

Thus we need an improvement of the verification by adding to the procedure that the final result will be sustained until it is found that the succeeding verifications have no Q_{ij}/m_i values which exceed the Q_{ij}/M_i value obtained by the preceding verifications.

8.3.3 Analysis of the Verification Applied to the Model 5

Let us see the case of model 5 as an example of the verification giving the incorrect result that the scene object numbered 5 corresponds to the model 5. The desired correct result is that there is no object corresponding to the model numbered 5. Let us consider why this incorrect result occurs.

The result is given by the second verification where the value of Q_{ij}/M_i, 0.67, exceeds the threshold value 0.4. However, we can see that the D_{ij} value of 681 is considerably higher than those of the other cases where correct results are obtained. The models of 1, 2, 3, 4, 6, 7, 8, 9 give the values of D_{ij}, 440, 209, 131, 485, 80, 204, 32, 337, respectively. This situation suggests that we need another kind of threshold which is used in the verification process concerning the distance value of D_{ij}. The new threshold is used in order to decide whether or not there is a model to be identified. The additional procedure is made such that the verification process will have no decision if the value of D_{ij} exceeds the threshold value concerning D_{ij}. In this example, if the threshold value is set to 500, any verification has no decision even if the value of Q_{ij}/M_i is large. This process will lead to a correct result for the recognition of the model 5 without any side effects for the other models.

8.3.4 Improvements of the Verification Algorithm

The detailed consideration by means of the three examples suggests that some improvements are necessary to obtain a robust verification algorithm. It is easy to embed the aspects indicated by the discussions concerning the examples of the models 4 and 5 into the verification algorithm. However, the aspect indicated by the example of the model 3 requires a major change of the verification algorithm. We will discuss the possibility of the major change in the next section.

8.4 The Weaker Condition for Determining Matched Facet Pairs

Since the incorrect recognition of the model 3 is basically caused by the splitting of the facet in the extraction process, the purpose of further improvement of the verification algorithm is to compensate splitting of the facet. One of the reasonable methods is to use a weak condition when we make matching facet pairs by producing the intersection part of L_{i_ℓ, j_ℓ}'s. In the experiment, we use the three largest L_{i_ℓ, j_ℓ}'s. If we use a set (defined by K^2) composed of two L_{i_ℓ, j_ℓ}'s, the intersection part will become larger and also will be able to include facet pairs which indicate weaker matching because of increase of ambiguity. Let us examine this situation by means of the example of model 3 (see Fig. 20).

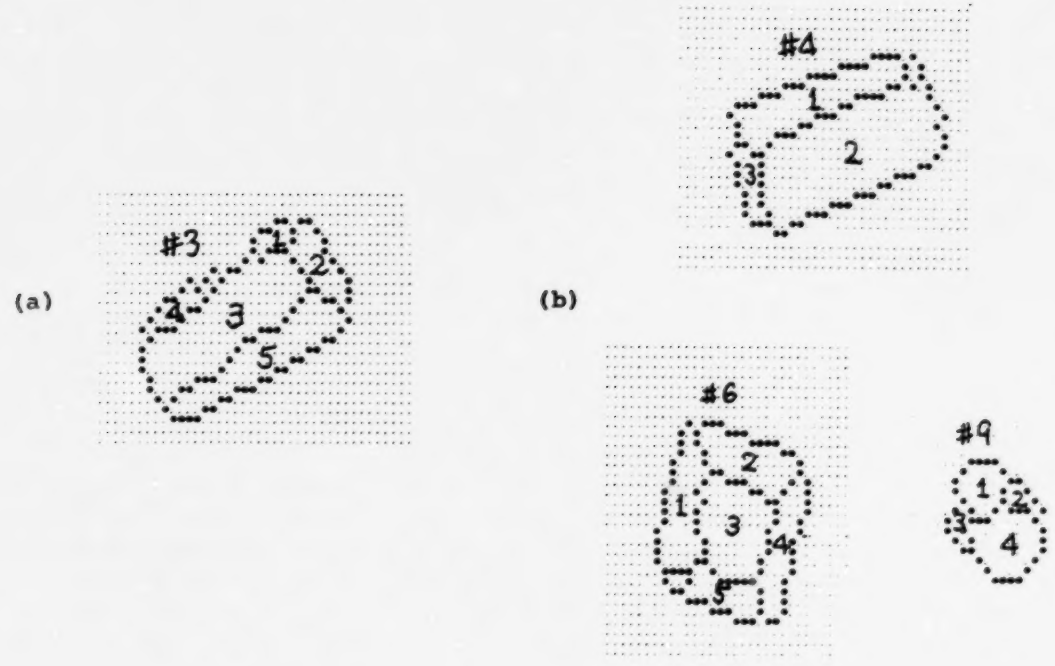


Figure 20. The model numbered 3 (a) (a part of Fig. 17) and scene blobs numbered 4, 9, and 6; (b) (parts of Fig. 18).

The first verification is carried out for the facet pair (2, 3), where the facet 2 and the facet 3 of the pair belong to the model 3 and the scene blob 4, respectively. The first verification process is based on the data as follows:

Model number=3	Scene number=4
Facet pair as brother of father ------	Children facet pairs
(model,scene) = (2,3)----------	{(2,3),(3,2),(4,2),(5,1)}
(model,scene) = (3,2)----------	{(3,2)}
(model,scene) = (4,2)----------	{(4,2)}
(model,scene) = (5,1)----------	{(2,3),(5,1)}

Then we obtain the parameters produced by the verification as follows:

$$(i_1,j_1) = (2,3), \quad L_{i_1,j_1} = \{2,5\}$$

$$(i_2,j_2) = (3,2), \quad L_{i_2,j_2} = \{3\}$$

$$(i_3,j_3) = (4,2), \quad L_{i_3,j_3} = \{4\}$$

$$(i_4,j_4) = (5,1), \quad L_{i_4,j_4} = \{2,5\}$$

Let us assume that we use the two largest L_{i_ℓ,j_ℓ}'s. Then the two L_{i_ℓ,j_ℓ}'s become L_{i_1,j_1} and L_{i_4,j_4}. Thus, we obtain

$$K^2 = \{(2,3), (5,2)\}$$

$$\bigcap_{\ell \in K^3} L_{i_\ell,j_\ell} = \{2,5\}$$

$$Q_{2,3} = 2, \; Q_{2,3}/M_3 = 0.40$$

The second verification is applied to the facet pair (3, 4) composed of the model 3 and the scene blob 9 with the brother pairs of (2, 3), (3, 4), (5, 1) and their children of $\{(2, 3)\}$, $\{(2, 3), (3, 4), (5, 1)\}$, $\{(5, 1)\}$, respectively. In this case, we have

$$K^2 = \{(2, 3), (3, 4)\} \text{ or } \{(3, 4), (5, 1)\}$$

$$\bigcap_{\ell \in K^3} L_{i_j,j_\ell} = \{2\} \text{ or } \{5\}$$

$$Q_{3,4} = 1, \; Q_{3,4n} = 0.20$$

The third verification is applied to the facet pair (5, 3) composed of the model 3 and the scene blob 6 with the brother pairs of (2, 5), (3, 4), (5, 2), (5, 3) and their children of $\{(1, 1), (2, 5)\}$, $\{(2, 5), (3, 4), (5, 2), (5, 3)\}$, $\{(5, 2)\}$, $\{(3, 1), (3, 2), (3, 4), (3, 5), (5, 3)\}$, respectively. In this case, we have

$$K^2 = \{(3,4), (5,3)\}$$

$$\bigcap_{\ell \in K^3} L_{i_\ell,j_\ell} = \{3,5\}$$

$$Q_{5,3}=2, \; Q_{5,3}/M_3 = 0.40$$

The verification will lead to a correct result that the scene blob numbered 4 corresponds to the model numbered 3.

The condition using K^2 is regarded as a weaker one compared with that of K^3. The weak condition can produce the facet pairs representing weak matching between facets. However, the weak condition also produces ambiguity of matched facet pairs. There is a suitable trade-off between the weakness of the matching condition and the robustness for covering splitting of facets.

9. DIRECTIONS FOR FUTURE RESEARCH

It is obvious that this report does not describe some important characteristics and related themes concerning the Cross-Angle Transform. Let us discuss aspects of two major problems in the following which should be considered in detail.

9.1 Mathematical Analysis of Ambiguity Reduction

Ambiguity reduction characteristics performs a key role when we apply the Cross-Angle Transform to the real problem of 3-D object recognition, since viewer-independent property is obtained by producing two kinds of freedom and at the same time the freedom yields ambiguities concerning relative normals and locations of facets.

It is important to reveal the mathematical structure of ambiguity reduction from the viewpoint of what kinds of conditions are required to eliminate ambiguity completely. In Chapter 4, we noted that the ambiguity concerning a normal vector is eliminated completely by using three facet pairs. However, the ambiguity concerning relative locations of facets is only reduced to a maximum of eight points when we use three facet pairs under no conditions. We must reveal some necessary and sufficient conditions which are intrinsic for recognizing of real 3-D objects.

9.2 Verification by the Use of Connectivity and Shape of Facets

In real recognition problems of 3-D objects, it is sometimes possible to extract other kinds of facet features such as connectivity and shape of a facet from range data. Despite that, connectivity and shape of the facet are not used for the Cross-Angle Transform; they are very useful for carrying out more effective verification for the result of primary matching.

First, let us consider how the connectivity feature is useful in the process of verification. The connectivity is usually defined among neighbourhood facets. The connectivity characteristics among facets of the model 1 and the scene 5 are described in Table 7.

Table 7. The connectivity matrices. Table 7a gives the matrix for the facets of the model number 1 (Fig. 17). Table 7b gives the matrix of the facets of the scene blob number 5 (Fig. 18). The number 1 of the matrix element indicates the existance of the connectivity and the number 0 indicates no connectivity.

	1	2	3	4	5	6	7	8
1	1	1	1	0	0	0	0	0
2	1	1	1	0	0	0	0	0
3	1	1	1	1	1	1	0	0
4	0	0	1	1	1	0	1	1
5	0	0	1	1	1	1	0	1
6	0	0	1	0	1	1	0	1
7	0	0	0	1	0	0	1	1
8	0	0	0	1	1	1	1	1

(a)

	1	2	3	4	5	6	7	8	9	10	11
1	1	1	1	1	0	0	0	0	0	0	0
2	1	1	1	0	1	0	0	0	0	0	0
3	1	1	1	1	1	1	1	1	1	1	0
4	1	0	1	1	0	0	1	0	0	0	0
5	0	1	1	0	1	1	0	0	0	0	0
6	0	0	1	0	1	1	0	0	0	0	0
7	0	0	1	1	0	0	1	0	0	1	0
8	0	0	1	0	0	1	0	1	1	0	0
9	0	0	1	0	0	0	0	1	1	1	0
10	0	0	1	0	0	0	1	0	1	1	1
11	0	0	0	0	0	0	0	0	0	1	1

(b)

In Chapter 8 we described that the first verification for recognizing the model 1 is applied to the father pair (8, 10) composed of the 8th facet of model 1 and the 10th facet of scene 5. Then the brother facet pairs of the father pair are listed from Table 4 as follows in Table 8.

It is reasonable to reduce the number of facet pairs by applying the connectivity characteristics to brother facet pairs in order to simplify the further verification process.

Let us define the so-called "consistency loop" by using both the connectivity relation and matching hypotheses between facets. Each of the brother facet pairs is regarded as a matching hypothesis. Let us consider two brother facet pairs in the list of brother facet pairs of Table 8. They are regarded as two matching hypotheses, that is,

hypothesis 1: model facet 1 -------- scene facet 9
hypothesis 2: model facet 3 -------- scene facet 1

If we define the relation (C_{o_2}) between model facet 1 and model facet 3 by using the matrix element of Table 7a, and also define the relation (C_{o_2}) between scene facet 9 and scene facet 1 by using Table 7(b), then the consistency loop is defined by the link as shown in Fig. 21.

Table 8. The brother facet pairs of the father (8, 10) composed of 8th facet of the model 1 and 10th facet of the scene 5.

Model facet =	1	2	2	2	2	3	4	5	6	6	6	8
Scene facet =	9	2	5	6	8	1	9	3	4	7	11	10

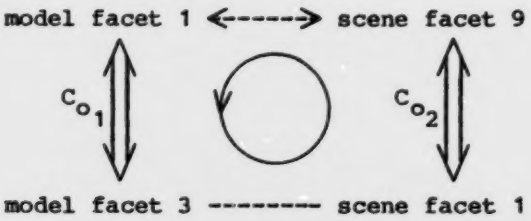


Figure 21. The definition of consistency loop.

Then we define that consistency loop as **complete** if $C_{o_1} = C_{o_2}$ holds, otherwise it is **incomplete**. The complete consistency loop means that hypothesis 1 is consistent with hypothesis 2. Moreover, we postulate that hypothesis 1 with connectivity value 0 is eliminated from the set of brother facet pairs when the consistency loop is incomplete. In this example, the hypothesis 1 is eliminated because C_{o_1} (=1) does not equal C_{o_2} (=0). If we repeat this elimination process for the brother facet pairs of Table 8, then we can obtain the new brother facet pairs as shown in Table 9.

It is obvious that the new brother facet pairs indicate a more exact matching result. This means that embedding of connectivity characteristics is useful in order to eliminate false matching facet pairs.

The same algorithm is applicable for children facet pairs produced by brother facet pairs in the further stage of verification.

If a shape feature is obtained by processing range data, we can use it as well as connectivity characteristics to eliminate false matching pairs. The more detailed discussion will be revealed in further research.

Table 9. The new brother facet pairs for Table 8 by eliminating false facet pairs using connectivity characteristics.

Model facet =	2	3	4	5	6	6	8
Scene facet =	2	1	9	3	4	7	10

9.3 Occlusion Problems

Another important experiment is to apply the algorithm to the scene including occluded objects from a view angle. This experiment will yield so-called occlusion problems. Occlusion problems should be a very good research target under the scheme of the Cross-Angle Transform. In the case of the occlusion problem, facets of an object obtained by range image data are usually split into parts by overlapped facets of other objects. A 3-D model description must be created by using data of multiple views and also by the method described in Chapter 5. The treatment of splitting facets will become essential. The situation seems to be more complex and difficult when we consider one-to-many correspondence among facets in order to treat split facets. However, we can see that the Cross-Angle Transform is quite suitable to treat one-to-many correspondence, as shown in Table 8. In this example, the facet 2 of the model 1 has four facets which are possible to match with. In fact, this situation is the reflection of splitting of a facet. The Cross-Angle Transform includes fundamentally the mechanism for treatment of one-to-many matching. A fundamental feature of the occlusion problem, also, is production of split facets. Therefore, the algorithm required in occlusion problems will to be similar to the one for treatment of splitting facets made by preprocessing of range data.

Thus we can believe that the Cross-Angle Transform will be applicable to real problems for recognizing occluded 3-D objects.

10. CONCLUSIONS

The Cross-Angle Transform is a method to obtain a structural description of 3-D objects which yields viewer-independent and robust characteristics. A single set of cross-angles obtained by the Cross-Angle Transform has two degrees of freedom to reconstruct completely the original 3-D object. The ambiguity caused by the degrees of freedom can be eliminated or considerably reduced by using three sets of cross-angles. Matching between a model and a blob to be recognized is carried out by calculating a distance for each pair of facets. The distance is defined between 2-D histograms each of which is produced with respect to a set of cross-angle pairs. In order to verify facet correspondences obtained by the matching, the verification process is contructed by using ambiguity reduction characteristics of the Cross-Angle Transform. The recognition experiments lead us to believe that this method is applicable to real 3-D scenes.

ACKNOWLEDGEMENTS

This work is carried out under the joint project on "Computer Vision and Image Processing" between the National Research Council of Canada and the Electrotechnical Laboratory of Japan. We would like to express thanks to the director of the Division of Electrical Engineering, National Research Council Canada, Dr. J.Y. Wong, the section head, Mr. N. Burtnyk, and the staff of the Computer Graphics Section for supporting the project and our work in Canada. We also would like to thank the director of the Electrotechnical Laboratory, Dr. K. Sato, the head of the Information Sciences Division, Dr. T. Nakajima, and the head of the Pattern Processing Section, Dr. N. Funakubo, for supporting the project in Japan.

REFERENCES

1. Kasvand, T., Oka, R., Dhome M., and Rioux, M. Segmentation, learning, and recognition of objects in range camera images. Proc. 4th Canadian CAD/CAM and Robotics Conference, Toronto, Ont. June 18-20, 1985.

2. Ballard, D.H. Generalizing the Hough transform to detect arbitrary shapes. Pattern Recognition, 13(20): 111-122; 1981.

3. Bajcsy, R. Three-dimensional scene analysis. Proc. 5th International Conference on Pattern Recognition, Miami, FL. 1980. pp. 1074-1082.

4. Bolles, R.C. Robust feature matching through maximum cliques. SPIE Technical Symposium on Imaging Applications for Automated Industrial Inspection and Assembly. 1979.

5. Ballard, D.H., and Sabbah, D. Viewer independent shape recognition. IEEE Trans. Pattern Anal. Mach. Intell. PAMI-5(6): 653-659; 1983.

6. Jarvis, R.A. A perspective on range finding techniques for computer vision. IEEE Trans. Pattern Anal. Mach. Intell. PAMI-5(2): 122-139; 1983.

7. Kanade, T., and Sommer, T.S. An optimal proximity sensor for measuring surface position and orientation for robot manipulation. Proc. SPIE, 449: 667-674; 1983.

8. Nitzan D., Rosen C., Agin G., Bolles R., Gleason, G., Hill, J., McGie, D., Prajoux, R., Park, W., Sword, A. Machine Intelligence Research Applied to Industrial Automation, Ninth Report, SRI International, Stanford, CA. August 1979.

9. Duda, R.O., Nitzan, D., and Barret, P. Use of range and reflectance data to find planar surface regions. IEEE Trans. Pattern Anal. Mach. Intell. PAMI-1(3): 259-271; 1979.

10. Inokuchi, S., Nita, T., Matsuda, F., and Sakurai, Y. A Three-dimensional edge-region operator for range pictures. 6th International Conference on Pattern Recognition, Munich. October 19-22, 1982. pp. 918-920.

11. Mitiche, A., and Aggarwal, J.K. Detection of edges using range information. IEEE Trans. Pattern Anal. Mach. Intell. PAMI-5(2): 174-178; 1983.

12. Morgenthaler, D.G., and Rosenfeld, A., Surfaces in three-dimensional digital images. Inf. Control, 51: 227-247; 1981.

13. Popplestone, R.J., Brown, C.M., Ambler, A.P., and Crawford, G.F. Forming models of plane-and-cylinder faceted bodies from light stripes. International Joint Conference on Artificial Intelligence. 1975.

14. Shirai, Y., and Suwa, M. Recognition of polyhedrons with a range finder. International Joint Conference on Artificial Intelligence. 1971. pp. 80-87.

15. Will, P.M., and Pennington, K.S. A preprocessing technique for robot and machine vision. International Joint Conference on Artificial Intelligence. 1971. pp. 66-70.

16. Davis, L.S. Shape matching using relaxation techniques. IEEE Trans. Pattern Anal. Mach. Intell. PAMI-1(1): 60-72; 1979.

17. Duda, O.R., and Hart, P.E. Pattern Classification and Scene Analysis. John Wiley and Sons, New York. 1973.

18. Guzman, A. Decomposition of a visual scene into three-dimensional bodies. Proc. AFIPS 1968 Fall Joint Computer Conference. pp. 291-304.

19. Mackworth, A.K. Interpreting pictures of polyhedral scenes. Artif. Intell. 4(2): 121-137; 1973.

20. Mackworth, A.K. Model-driven interpretation in intelligent vision systems. Perception, 5: 349-370; 1976.

21. Oka, R., Kasvand, T., and Rioux, M. Cross-angle transform for viewer-independent recognition of 3-D objects. Proc. 1985 Computer Vision and Pattern Recognition, San Francisco, CA. June 9-13, 1985. pp. 470-475.

22. Oka, R. Ambiguity reduction of the cross-angle transform for 3-D objects recognition. In press.

23. Oshima, M., and Shirai, Y. Object recognition using three-dimensional information. IEEE Trans. Pattern Anal. Mach. Intell. PAMI-5(4): 353-361; 1983.

24. Sugihara, K. Range-data analysis guided by a junction dictionary. Artif. Intell. 12: 41-69; 1979.

25. Kanade, T. A Theory of Origami World. Technical Report CMU-CS-78-144, Computer Science, Carnegie-Mellon University, Pittsburg, PA. 1978.

26. Rioux, M. Laser range finder based on synchronized scanners. Appl. Opt. 23(21): 3837-3844; 1984.

27. Haralick, R.M. Edge and region analysis for digital image data. Comput. Graphics Image Process. 12: 60-73; 1980.

28. Jambu, M., and Lebeaux, M.O. Cluster Analysis and Data Analysis. North-Holland Publishing Company. 1983.

29. Kasvand, T., and Otsu, N. Segmentation of thinned binary scenes with good connectivity algorithms. 7th International Conference on Pattern Recognition, Montreal, P.Q. July-Aug. 1984. pp. 297-300.

APPENDIX A: RELATION BETWEEN A GRADIENT VECTOR AND A NORMAL VECTOR

Let us assume that a normal vector is (a, b, c). Then the plane on which the normal vector is located is described by:

$$aX + bY + cZ = d$$

Partial differential of Z(X, Y) leads to

$$\frac{\partial Z}{\partial X} = -\frac{a}{c} = p, \qquad \frac{\partial Z}{\partial Y} = -\frac{b}{c} = q$$

where (p, q) is a gradient vector. On the other hand, the geometrical analysis as shown in Fig. A-1 leads to the relation:

$$(a,b,c) = (-\cos\psi \cdot \cos\theta, -\cos\psi \cdot \sin\theta, \sin\psi)$$

where

$$\tan\theta = \frac{q}{p}, \qquad (0 \leq \theta \leq 2\pi)$$

Then we have the relations as follows:

$$p = -\frac{a}{c} = -\frac{\cos\psi \cdot \cos(\theta + \pi)}{\sin\psi} = \frac{\cos\theta}{\tan\psi}$$

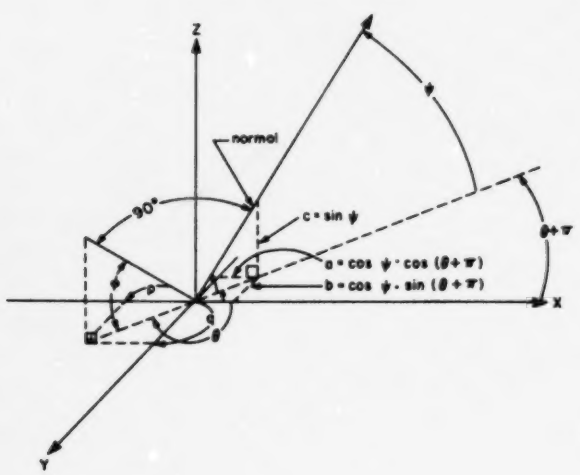


Figure A-1. The notations of the normal vector (a, b, c), the slant angle, and the tilt angle

$$q = -\frac{b}{c} = -\frac{\cos\psi \cdot \sin(\theta + \pi)}{\sin\psi} = \frac{\sin\theta}{\tan\psi}$$

$$p^2 + q^2 = \frac{1}{\tan^2\psi}$$

$$\tan\psi = \pm\frac{1}{(p + q)^{1/2}} = \tan\left(\frac{\pi}{2} - \phi\right) = \cot\phi$$

$$\tan\phi = \pm(p + q)^{1/2}$$

$$\phi = \begin{cases} \phi_1 \\ \phi_1 + \pi \end{cases} \qquad (0 < \phi_1 < \frac{\pi}{2})$$

$$\psi = \frac{\pi}{2} - \phi = \begin{cases} \psi_1: & 0 < \psi_1 < \frac{\pi}{2} \\ \psi_2: & -\pi < \psi_2 < \frac{\pi}{2} \end{cases}$$

ψ_1: outer direction of the normal vector

ψ_2: inner direction of the normal vector

APPENDIX B: ROTATION EQUALIZATION OF AN OBJECT USING TWO COMMON FACETS OF DIFFERENT VIEWS

Let us consider two sets of facets produced by two different views of the same object. Let us assume that the first facet set includes the two facets identified by the symbol A and B, and the facets A and B are corresponding to the facet A' and B' of the second facet set, respectively. Each facet has two kinds of parameters, the normal vector and the location in 3-D space. Then we can consider the problem of shifting and rotation in 3-D space in order to overlap A' and B' facets with A and B facets, respectively. The union of the two facet sets is desirable for representing more precise three-dimensional structure of the object. It is obvious that the union is obtained using the first facet set and the second facet set whose facets are applied by the shifting and rotation transform for overlapping A' and B' facets with A and B facets at the same time, since a simple geometrical analysis leads to the conclusion that facets of two facet sets can occupy the same space in 3-D space when at least two facets are matching each other between them.

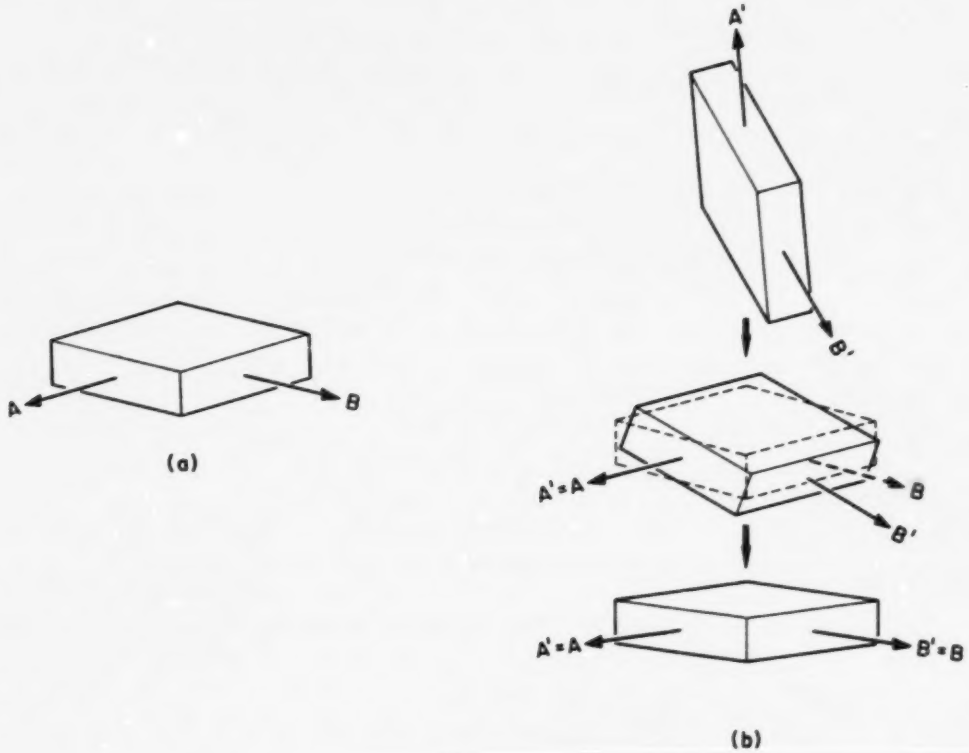


Figure B-1. The schematic explanation of the transform for obtaining the union of the first and the second angle facet sets.

Let us formulate the transform in the following. Figure B-1 shows the schematic explanation of the transform. The transform is composed of (1) matching between the normal vectors of A facet and A' facet and (2) the matching between the locations of B facet and B' facet.

Let us assume that (a, b, c), (0, 0, 0) are the normal vector and the coordinate of the facet A, respectively. The (a', b', c'), (A'_x, A'_y, A'_z) are for the facet A', (d, e, f), (B_x, B_y, B_z) for the facet B, and (d', e', f'), (B'_x, B'_y, B'_z) for the facet B'. Then we can obtain the following relations about the normal (a, b, c).

$$a = \cos\psi \cdot \sin\theta$$
$$b = \cos\psi \cdot \cos\theta$$
$$c = \sin\psi$$
$$\theta = \arctan a/b$$
$$\psi = \arctan c/(a + b)^{1/2}$$

where ψ and θ are indicated in the Fig. B-2.

If the facet set of the second object is composed of the facets, namely, $\{F_i = ((\ell_i, m_i, n_i), (X_i, Y_i, Z_i)) : i = 1, 2, \ldots, K)\}$, then the first procedure for matching the normals of A and A' facets is shifting of (X_i, Y_i, Z_i), $i = 1, 2, \ldots, K$ such that:

$$(\underline{X}_i, \underline{Y}_i, \underline{Z}_i) = (X_i, Y_i, Z_i) - (A'_x, A'_y, A'_z),\ i = 1, 2, \ldots, K,$$

where $(\underline{X}_i, \underline{Y}_i, \underline{Z}_i)$ indicates the shifted coordinate of (X_i, Y_i, Z_i).

The second procedure is the rotation of the normals of all the facets included in the second set. The rotation needed is defined by

$$\Delta\psi = \psi - \psi'$$

$$\Delta\theta = \theta - \theta'$$

where ψ', θ' are the angles with respect to the normal (a', b', c') of A'.

Then the rotated normal of the facet F_i with ψ_i and θ_i is given by:

$$(\ell_I, m_I, n_I) = (\cos(\psi_i + \Delta\psi)\sin(\theta_i + \Delta\theta),\ \cos(\psi_i + \Delta\psi)\cos(\theta_i + \Delta\theta),\ \sin(\psi_i + \Delta\psi))$$

and the new location (X_I, Y_I, Z_I) is also given by:

$$\begin{pmatrix} X_I \\ Y_I \\ Z_I \end{pmatrix} = \begin{pmatrix} \cos\Delta\psi \cos\Delta\theta & \cos\Delta\psi \sin\Delta\theta & -\sin\Delta\psi \\ -\sin\Delta\theta & \cos\Delta\theta & 0 \\ \sin\Delta\psi \cos\Delta\theta & \sin\Delta\psi \sin\Delta\theta & \cos\Delta\psi \end{pmatrix} \begin{pmatrix} \underline{X}_i \\ \underline{Y}_i \\ \underline{Z}_i \end{pmatrix}$$

In this stage, the location (B_x, B_y, B_z) of the first object does not yet match with the new location (B''_x, B''_y, B''_z) obtained by the shift and rotation of (B'_x, B'_y, B'_z). It is easy to obtain the angles (ψ_1, θ_1) and (ψ_2, θ_2) with respect to (B_x, B_y, B_z) and (B''_x, B''_y, B''_z) as shown in Fig. B-3.

The rotation needed in order to match (B''_x, B''_y, B''_z) with (B_y, B_y, B_z) is defined by:

$$\Delta\tilde{\psi} = \psi_1 - \psi_2$$

$$\Delta\tilde{\theta} = \theta_1 - \theta_2$$

The final procedure is the rotation of the second facet set $\{F_i = ((\ell_I, m_I, n_I), (X_I, Y_I, Z_I)): I = 1, 2, \ldots, K\}$ by the difference quantities $\Delta\tilde{\psi}$ and $\Delta\tilde{\theta}$ by the same way as mentioned before.

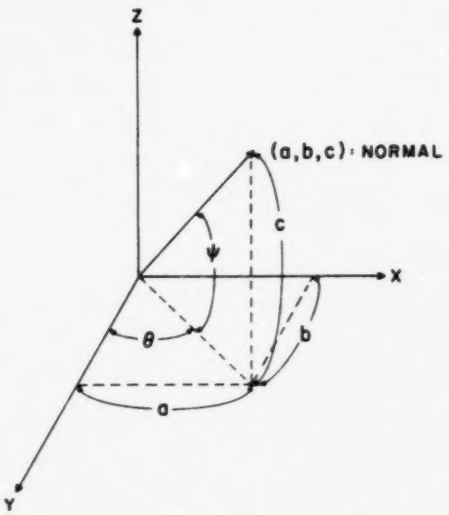


Figure B-2. The notations of the normal vector (a, b, c) and the angles ψ and θ.

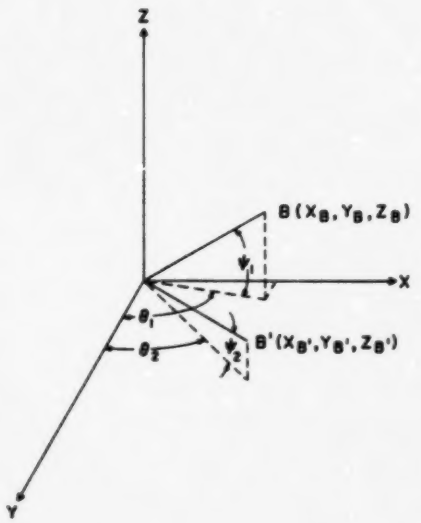


Figure B-3. The notations of (B_x, B_y, B_z), (B''_x, B''_y, B''_z), ψ_1, ψ_2, θ_1, θ_2.

The union facet of the first set and second set is obtained by adding the first facet set to the facets of the second set which satisfy the condition such that

$$(\underline{X}_i - \underline{X}_j)^2 + (\underline{Y}_i - \underline{Y}_j)^2 + (Z_i - \underline{Z}_j)^2 > H, \text{ for all } i$$

where $(\underline{X}_i, \underline{Y}_i, \underline{Z}_i)$ and $(\underline{X}_j, \underline{Y}_j, \underline{Z}_j)$ indicate the location of ith facet of the first set and the tranformed location of jth facet of the second set, respectively, and H is a given threshold value.

APPENDIX C: PROGRAMS FOR 3-D OBJECT RECOGNITION

```
C
C
CCCC MAIN PROGRAM OF 3DSYS.FOR JAN.18, 1985
      DIMENSION IAA(128,128,5),ATT(111,9),IZB(128),MBUF(40,2),IBUF(40,2)
      DIMENSION JAA(128,128,5),BTT(111,9)
      INTEGER*2 CFN(11,40,2),NFN(11,40,40,2),MSLCT(11,3),INSLCT(11,11,3)
      INTEGER*2 DIS(11,11,3)
      INTEGER MSG(11),ABMDL(20,16,16,11),MSP(20,11)
      INTEGER*2 MABM(20,20,2,11)
      DIMENSION AMM(20,11),BMM(20,11)
      INTEGER ISG(11),ALBT(20,16,16,11),ISP(20,11)
      INTEGER*2 IABI(20,20,2,11)
      DIMENSION AII(20,11)
      DIMENSION MDL(9,4,2)
      INTEGER*2 MMS(3,11,3,30)
C
C FILE NO=31 --- MMM1.DAT FOR 3-D PICTURE
C FILE NO=32 --- MMM2.DAT FOR 3-D PICTURE
C FILE NO=33 --- MMM3.DAT FOR 3-D PICTURE
C FILE NO=34 --- MMM4.DAT FOR 3-D PICTURE
C FILE NO=35 --- MMM5.DAT FOR 3-D PICTURE
C FILE NO=36 --- MMM6.DAT FOR 3-D PICTURE
C
C INTEGER*4 FOR FILE 41-47
C
C IAA(IX,IY,1) ; DISTANCE 1MM=100 UNIT
C IAA(IX,IY,2) ; BLOB (EACH PIXEL OF A BLOB HAS SAME LABEL)
C IAA(IX,IY,3) ; PROJECTED ANGLE OF NOMAL ON X-Y PLANE (THETA)
C IAA(IX,IY,4) ; NORNAL ANGLE (THETA-HAT)
C IAA(IX,IY,5) ; SURFACE LABEL
C                (#109, BOUNDARY=1,REGION=2,3,...
C                       NO ZERO (NO SPACE PIXEL))
C
C ATT ---- REAL NUMBER
C
C ATT(IG,1) ; AVERAGE OF X
C ATT(IG,2) ; AVERAGE OF Y
C ATT(IG,3) ; AVERAGE OF Z
C ATT(IG,4) ; X COMPONENT OF THE NORMAL VECTOR
C ATT(IG,5) ; Y COMPONENT OF THE NORMAL VECTOR
C ATT(IG,6) ; Z COMPONENT OF THE NORMAL VECTOR
C ATT(IG,7) ; 3-D AREA OF THE FACET
C
```

```
C MB          ; BLOB NUMBER IN SCENE
C I AND J     ; LOCAL FACET NUMBER IN BLOB
C IG          ; GLOBAL FACET NUMBER IN SCENE
C
C ABMDL(I,ALPHA,BETA,MB) AND ALBT(I,ALPHA,BETE,MB)
C
C           ALPHA ... CROSSING ANGLE BETWEEN I-TH NORMAL
C                     VECTOR AND OTHER NORMAL
C           BETA  ... CROSSING ANGLE BETWEEN I-TH NORMAL
C                     VECTOR AND THE VECTOR FROM THE CENTER
C                     POINT OF I-TH FACET TO THAT OF
C                     ANOTHER FACET
C
C ALBT IS THE 2-DIMENSIONAL HISTOGRAM OF AREA
C
C MSP/ISP(I,MB).........GLOBAL FACET NUMBER FOR EACH BLOB
C MABM/IABI(I,J,1,MB)...QUANTIZED ALPHA
C MABM/IABI(I,J,2,MB)...QUANTIZED BETA
C MSG/ISG(MB)...........TOTAL FACET NUMBER OF EACH BLOB
C AMM/AII(I,MB).........REAL FACET AREA OF EACH BLOB
C
 9999 CONTINUE
      PRINT *,'RESLULT PRINT : DISPLAY=3, LP=10'
      READ(5,*) IOUT
CC    PRINT *,'KA AND KB='
CC    READ(5,*) KA,KB
      KA=8
      KB=7
      WRITE(IOUT,2001) KA,KB
 2001 FORMAT(1H ,'ALPHA QUANTIZED BY ',I2,' BETA QUANTIZED BY ',I2)
      WRITE(IOUT,2002)
 2002 FORMAT(1H )
C
C
C
      DO 2000 IK=1,5
      KK=40+IK
      DO 1000 J=1,128
      READ(KK) (IZB(JK),JK=1,128)
      DO 1001 I=1,128
 1001 IAA(I,J,IK)=IZB(I)
 1000 CONTINUE
      REWIND(KK)
 2000 CONTINUE
      CALL REGION(IAA,ATT,MSEG,MBL,MSP,MSG,AMM)
```

```
      CALL CATH(MBL,ATT,MSG,MSP,KA,KB,ABMDL,MABM)
CC    CALL OUT1(IOUT,MBL,MSG,MSP,ATT)
CC    CALL OUT2(IOUT,MBL,MSG,MABM)
      DO 2500 IK=1,5
      KK=45+IK
      DO 1500 J=1,128
      READ(KK) (IZB(JK),JK=1,128)
      DO 1501 I=1,128
 1501 IAA(I,J,IK)=IZB(I)
 1500 CONTINUE
      REWIND(KK)
 2500 CONTINUE
      CALL REGION(IAA,BTT,ISEG,IBL,ISP,ISG,AII)
      CALL CATH(IBL,BTT,ISG,ISP,KA,KB,ALBT,IABI)
CC    WRITE(IOUT,1552)
CC    CALL OUT1(IOUT,IBL,ISG,ISP,BTT)
CC    CALL OUT2(IOUT,IBL,ISG,IABI)
      CALL MATCH(IBL,ALBT,ISG,ISP,IABI,MBL,ABMDL,MSG,MSP,MABM,
     &AMM,KA,KB,DIS,MSLCT,INSLCT,IOUT)
C
      DO 3000 ICAN=1,3
      NCAN=ICAN
      CALL CORRES(DIS,MSLCT,INSLCT,ISP,MSP,IABI,MABM,
     &ALBT,ABMDL,MSG,ISG,IBL,MBL,CFN,NFN,NCAN,IOUT)
CC    WRITE(IOUT,3210)
 3210 FORMAT(1H1)
CC    WRITE(IOUT,4000) NCAN
 4000 FORMAT(1H ,'ORDER OF THE CANDIDATE=',I2)
CC    WRITE(IOUT,1559)
      DO 1400 I=1,MBL
      DO 1440 IW=1,40
      DO 1440 IK=1,2
      MBUF(IW,IK)=0
 1440 IBUF(IW,IK)=0
      II=CFN(I,2,1)
      DO 1450 KK=4,40
      IF(CFN(I,KK,1).NE.0) THEN
      DO 1451 K=1,2
 1451 MBUF(KK,K)=MABM(II,CFN(I,KK,1),K,I)
      ENDIF
      IF(CFN(I,KK,2).NE.0) THEN
      JJ=CFN(I,2,2)
      JK=CFN(I,1,2)
      DO 1452 K=1,2
      IBUF(KK,K)=IABI(JJ,CFN(I,KK,2),K,JK)
```

```
 1452 CONTINUE
      ENDIF
 1450 CONTINUE
CCX   WRITE(IOUT,1552)
CCX   WRITE(IOUT,1550) I,(CFN(I,KK,1),KK=2,28)
CCX   WRITE(IOUT,1551)   (CFN(I,KK,2),KK=1,28)
CX    WRITE(IOUT,1554) ((IBUF(KK,K),K=1,2),KK=4,28)
 1400 CONTINUE
 1559 FORMAT(1H ,16X,' "FATHER" PAIR ',20X,' "BROTHER OF FATHER" PAIR')
 1550 FORMAT(1H ,'MODEL BLOB  =',I2,1X,'FACET= ',I2,
     &1X,'(DIS=',I4,')',3X,33I3)
 1551 FORMAT(1H ,'SCENE BLOB  =',I2,1X,'FACET= ',I2,
     &1X,'(DIS=',I4,')',3X33I3)
 1552 FORMAT(1H )
 1554 FORMAT(1H ,29X,'ALPHA BETA=',33(2I1,1X))
      CALL CHILD(CFN,NFN,MBL,IOUT,ICAN,MMS)
C
 3000 CONTINUE
C
      CALL FINAL(MBL,ATT,BTT,MSP,ISP,MSG,MMS,IOUT)
C
      PRINT *,'CONTINUE(0) NO(-1)'
      READ(5,*) ICONT
      IF(ICONT.EQ.0) GOTO 9999
 8765 CONTINUE
      STOP
      END
C
      SUBROUTINE REGION(IAA,ATT,ISEG,IBL,ISP,MSG,AMM)
      DIMENSION IAA(128,128,5),ATT(111,9),IQ(111)
      DIMENSION ISP(20,11),MSG(11),AMM(20,11),NZ(200),IS(200),IK(200)
      ISS=0
      IKK=0
      DO 2002 IX=1,128
      DO 2002 IY=1,128
      N2=IAA(IX,IY,2)
      N5=IAA(IX,IY,5)
      IF(N2.NE.0) ISS=MAX0(ISS,N2)
      IF(N5.GE.3) IKK=MAX0(IKK,N5)
      IF(N5.LE.2.OR.N2.EQ.0) IAA(IX,IY,5)=0
 2002 CONTINUE
      IBL=ISS
      ISEG=IKK
      DO 1000 I=1,111
      IQ(I)=0
```

```
      DO 1000 J=1,9
1000  ATT(I,J)=0.0
      DO 21 I=1,128
      DO 21 J=1,128
      INUM=IAA(I,J,5)
      IF(INUM.LE.0) GOTO 21
      IQ(INUM)=IQ(INUM)+1
      XI=FLOAT(I)
      YJ=FLOAT(J)
      ZZ=FLOAT(IAA(I,J,1))
      ZZ=ZZ/100.0
      AX=360.0-FLOAT(IAA(I,J,3))
      AX=(AX*3.1412)/180.0
      BX=FLOAT(IAA(I,J,4))*3.1412
      BX=BX/180.0
      ATT(INUM,1)=ATT(INUM,1)+XI
      ATT(INUM,2)=ATT(INUM,2)+YJ
      ATT(INUM,3)=ATT(INUM,3)+ZZ
      ATT(INUM,4)=ATT(INUM,4)+COS(BX)*COS(AX)
      ATT(INUM,5)=ATT(INUM,5)+COS(BX)*SIN(AX)
      ATT(INUM,6)=ATT(INUM,6)+SIN(BX)
      ATT(INUM,7)=ATT(INUM,7)+BX
   21 CONTINUE
      JJJ=0
      DO 22 II=1,111
      IF(IQ(II).LE.2) GOTO 22
      JJJ=JJJ+1
      QQ=FLOAT(IQ(II))
      DO 23 JJ=1,7
   23 ATT(JJJ,JJ)=ATT(II,JJ)/QQ
      CC=ATT(JJJ,7)
      ATT(JJJ,7)=QQ/SIN(CC)
   22 CONTINUE
      NNN=0
      DO 31 J=1,JJJ
      JZ=IFIX(ATT(J,3))
      IF(JZ.LE.5) THEN
      NNN=NNN+1
      NZ(NNN)=J
      ENDIF
   31 CONTINUE
      IF(NNN.EQ.0) THEN
      ISEG=JJJ
      GOTO 8888
      ELSE
```

```
      MM=0
      DO 331 J=1,JJJ
      DO 332 K=1,NNN
  332 IF(NZ(K).EQ.J) GOTO 331
      MM=MM+1
      DO 333 L=1,7
  333 ATT(MM,L)=ATT(J,L)
  331 CONTINUE
      ISEG=MM
      ENDIF
 8888 CONTINUE
      IZZ=1
      DO 34 I=1,200
      IN=0
      DO 33 J=1,ISEG
      JX=IFIX(ATT(J,1)+0.5)
      JY=IFIX(ATT(J,2)+0.5)
      IF(IAA(JX,JY,2).EQ.0) THEN
      PRINT *,'IAA(JX,JY.2).EQ.0'
CC    STOP
      ENDIF
      IF(IAA(JX,JY,2).EQ.I) THEN
      IN=IN+1
      IF(IN.GT.20) THEN
      PRINT *,'IN OVER 20'
      STOP
      ENDIF
      ISP(IN,IZZ)=J
      AMM(IN,IZZ)=ATT(J,7)
      ENDIF
   33 CONTINUE
      IF(IN.NE.0) THEN
      MSG(IZZ)=IN
      IZZ=IZZ+1
      ENDIF
   34 CONTINUE
      IBL=IZZ-1
      ILL=0
CC    PRINT *,'ISEG=',ISEG,'MSG(IL)=',(MSG(IL),IL=1,9)
      DO 40 IZ=1,IBL
   40 ILL=ILL+MSG(IZ)
      IF(ILL.NE.ISEG) THEN
      PRINT *,'ILL.NE.ISEG'
      STOP
      ENDIF
```

```
      RETURN
      END
C
      SUBROUTINE CATH(IBL,ATT,MSG,ISP,KA,KB,ALBT,KABK)
      DIMENSION ATT(111,9),ALBT(20,16,16,11),R(3),S(3)
      INTEGER ALBT,ISP(20,11),MSG(11)
      INTEGER*2 KABK(20,20,2,11)
      DO 4000 I=1,20
      DO 4000 J=1,16
      DO 4000 K=1,16
      DO 4000 L=1,11
 4000 ALBT(I,J,K,L)=0
      DO 4001 I=1,20
      DO 4001 J=1,20
      DO 4001 K=1,2
      DO 4001 L=1,11
 4001 KABK(I,J,K,L)=0
      DO 1200 MB=1,IBL
      DO 100 IP=1,MSG(MB)
      DO 200 JQ=1,MSG(MB)
      I=ISP(IP,MB)
      J=ISP(JQ,MB)
      ICON=0
      DO 1 K=1,3
      R(K)=ATT(I,3+K)
    1 S(K)=ATT(J,3+K)
 1000 CONTINUE
      RR=0.0
      DO 2 K=1,3
    2 RR=RR+R(K)*S(K)
      QQ=0.0
      QQ=QQ+(R(2)*S(3)-R(3)*S(2))**2
      QQ=QQ+(R(3)*S(1)-R(1)*S(3))**2
      QQ=QQ+(R(1)*S(2)-R(2)*S(1))**2
      QQ=SQRT(QQ)
      IF(QQ.GE.0.00000001.AND.ABS(RR).GE.0.00000001) THEN
      ANGLE=ATAN2(QQ,RR)
      ELSE
C     PRINT *,'ATAN2 IRREGULAR !!!!'
C     STOP
      ANGLE=0.0
      ENDIF
      IF(ICON.EQ.0) THEN
      ALP=ANGLE
      DO 3 K=1,3
```

```
    3 S(K)=ATT(J,K)-ATT(I,K)
      SS=S(1)**2+S(2)**2+S(3)**2
      SS=SQRT(SS)
      IF(ABS(SS).GE.0.00000001) THEN
      DO 4 K=1,3
    4 S(K)=S(K)/SS
      ELSE
      DO 44 K=1,3
   44 S(K)=0.0
      ENDIF
      ICON=ICON+1
      GOTO 1000
      ELSE
      BETA=ANGLE
      IF(ICON.EQ.1) GOTO 9
      ENDIF
    9 CONTINUE
      TA=FLOAT(KA)
      TB=FLOAT(KB)
      TA=3.1412/TA
      TB=3.1412/TB
      DO 10 K=1,16
      TA1=FLOAT(K-1)*TA
      TA2=FLOAT(K)*TA
      TB1=FLOAT(K-1)*TB
      TB2=FLOAT(K)*TB
      IF(K.EQ.KA) TA2=TA2+0.01
      IF(K.EQ.KB) TB2=TB2+0.01
      IF(ALP.GE.TA1.AND.ALP.LT.TA2) IA=K
      IF(BETA.GE.TB1.AND.BETA.LT.TB2) IB=K
   10 CONTINUE
      ALBT(IP,IA,IB,MB)=ALBT(IP,IA,IB,MB)+IFIX(ATT(J,7))
      KABK(IP,JQ,1,MB)=IA
      KABK(IP,JQ,2,MB)=IB
  200 CONTINUE
  100 CONTINUE
 1200 CONTINUE
      RETURN
      END
C
      SUBROUTINE MATCH(IBL,ALBT,ISG,ISP,IABI,MBL,ABMDL,MSG,MSP,MABM,
     &AMM,KA,KB,DIS,MSLCT,INSLCT,IOUT)
      INTEGER ALBT(20,16,16,11),ISG(11),ISP(20,11)
      INTEGER ABMDL(20,16,16,11),MSG(11),MSP(20,11)
      INTEGER*2 IABI(20,20,2,11),MABM(20,20,2,11)
```

```
      DIMENSION AMM(20,11)
      DIMENSION XAREA(111),DD(111),LNN(111),NAA(111)
      INTEGER*2 DIS(11,11,3),MSLCT(11,3)
      INTEGER*2 INSLCT(11,11,3)
      DO 4001 I=1,11
      DO 4001 J=1,11
      DO 4001 K=1,3
      DIS(I,J,K)=0
 4001 INSLCT(I,J,K)=0
      DO 1000 MD=1,MBL
      DO 500 I=1,MSG(MD)
      XAREA(I)=-AMM(I,MD)
  500 CONTINUE
      KX=MSG(MD)
      CALL XORDER(XAREA,NAA,KX,KX)
      DO 444 IW=1,3
      DO 443 IR=1,MSG(MD)
      IF(NAA(IR).EQ.IW) THEN
      MSLCT(MD,IW)=IR
      ENDIF
  443 CONTINUE
  444 CONTINUE
      DO 4000 ILL=1,3
      DO 100 I=1,20
  100 DD(I)=0
      DO 1 J=1,MSG(MD)
    1 IF(NAA(J).EQ.ILL) JJ=J
      DO 11 IC=1,IBL
      NNN=ISG(IC)
      DO 10 I=1,NNN
      ID=0
      DO 3 K=1,KA
      DO 4 L=1,KB
      IQSN=ALBT(I,K,L,IC)
      IQMM=ABMDL(JJ,K,L,MD)
      IDIF=IABS(IQSN-IQMM)
      IF(IQMM.EQ.0) IDIF=2*IDIF
      ID=ID+IDIF
    4 CONTINUE
    3 CONTINUE
      DD(I)=FLOAT(ID)
   10 CONTINUE
      CALL XORDER(DD,LNN,NNN,NNN)
      DO 300 I=1,NNN
      IF(LNN(I).EQ.1) THEN
```

```
      INSLCT(MD,IC,ILL)=I
      DIS(MD,IC,ILL)=IFIX(DD(I))
      ENDIF
  300 CONTINUE
   11 CONTINUE
 4000 CONTINUE
 1000 CONTINUE
CCX   WRITE(IOUT,3100)
 3100 FORMAT(1H1)
CCX   WRITE(IOUT,3001)
CCX   WRITE(IOUT,3002) (IK,IK=1,9)
 3002 FORMAT(1H ,'OBJECT BLOB=',6X,9(I3,6X))
CCX   WRITE(IOUT,3101)
 3101 FORMAT(1H ,16X,'(FCT,DIS)(FCT,DIS)(FCT,DIS)(FCT,DIS)(FCT,DIS)
     &(FCT,DIS)(FCT,DIS)(FCT,DIS)(FCT,DIS)')
CCX   WRITE(IOUT,3001)
      DO 400 I=1,MBL
      DO 401 IL=1,3
CCX   WRITE(IOUT,3000) I,MSLCT(I,IL),((INSLCT(I,IC,IL),
CCX  &DIS(I,IC,IL)),IC=1,IBL)
  401 CONTINUE
CCX   WRITE(IOUT,3001)
  400 CONTINUE
 3000 FORMAT(1H ,'MDL=',I2,1X,'FACET=',I2,1X,10('(',I2,',',I4,')'))
 3001 FORMAT(1H )
      RETURN
      END
C
      SUBROUTINE CORRES(DIS,MSLCT,INSLCT,ISP,MSP,IABI,MABM,
     &ALBT,ABMDL,MSG,ISG,IBL,MBL,CFN,NFN,NCAN,IOUT)
      INTEGER MSG(11),ABMDL(20,16,16,11),MSP(20,11)
      INTEGER*2 MABM(20,20,2,11)
      INTEGER ISG(11),ALBT(20,16,16,11),ISP(20,11)
      INTEGER*2 IABI(20,20,2,11)
      INTEGER*2 CFN(11,40,2),DIS(11,11,3),MSLCT(11,3)
      INTEGER*2 INSLCT(11,11,3),NFN(11,40,40,2)
      DIMENSION LNN(100),DD(100)
      DO 100 I=1,11
      DO 100 J=1,40
      DO 100 K=1,2
  100 CFN(I,J,K)=0
      DO 1000 I=1,MBL
      NIN=0
      DO 1 IC=1,IBL
      DO 1 IL=1,3
```

```
      NIN=NIN+1
    1 DD(NIN)=FLOAT(DIS(I,IC,IL))
C
C
      DO 1029 IC=1,IBL
      NNX=3*(IC-1)
      DQ=AMIN1(DD(NNX+1),DD(NNX+2),DD(NNX+3))
      IFLAG=0
      DO 1030 IL=1,3
      IF(ABS(DQ-DD(NNX+IL)).LE.0.0001.AND.IFLAG.EQ.0) THEN
      IFLAG=1
      GOTO 1030
      ELSE
      DD(NNX+IL)=10000.0
      ENDIF
 1030 CONTINUE
 1029 CONTINUE
C
      CALL XORDER(DD,LNN,NIN,NIN)
      DO 2 II=1,NIN
    2 IF(LNN(II).EQ.NCAN) IIX=II
      MIM=0
      DO 4 IC=1,IBL
      DO 4 IL=1,3
      MIM=MIM+1
      IF(MIM.EQ.IIX) THEN
      ICC=IC
      ILL=IL
      ENDIF
    4 CONTINUE
      JJ=INSLCT(I,ICC,ILL)
      II=MSLCT(I,ILL)
      CFN(I,2,1)=II
      CFN(I,3,1)=DIS(I,ICC,ILL)
      CFN(I,1,2)=ICC
      CFN(I,2,2)=JJ
      CFN(I,3,2)=DIS(I,ICC,ILL)
      KTK=3
      DO 5 J=1,MSG(I)
      KAX=MABM(II,J,1,I)
      KBX=MABM(II,J,2,I)
      IQML=ABMDL(II,KAX,KBX,I)
      IF(IQML.EQ.0) GOTO 5
      DO 6 JI=1,ISG(ICC)
      IAX=IABI(JJ,JI,1,ICC)
```

```
      IBX=IABI(JJ,JI,2,ICC)
      IF(IAX.EQ.KAX.AND.IBX.EQ.KBX) THEN
      KTK=KTK+1
      IF(KTK.GT.40) THEN
      STOP
      ENDIF
      CFN(I,KTK,2)=JI
      CFN(I,KTK,1)=J
      ENDIF
    6 CONTINUE
    5 CONTINUE

 1000 CONTINUE
C
C
      DO 2000 I=1,MBL
      ICC=CFN(I,1,2)
      DO 2001 IK=1,40
      DO 2001 K=1,40
      DO 2001 J=1,2
 2001 NFN(I,IK,K,J)=0
C
      DO 2002 K=4,40
      IF(CFN(I,K,1).EQ.0) GOTO 2002
      II=CFN(I,K,1)
      JJ=CFN(I,K,2)
      NFN(I,K,40,1)=II
      NFN(I,K,40,2)=JJ
C     PRINT *,'I=',I,' K=',K,' II=',II,' JJ=',JJ
      KKK=0
      DO 2005 J=1,MSG(I)
      KAX=MABM(II,J,1,I)
      KBX=MABM(II,J,2,I)
      IQML=ABMDL(II,KAX,KBX,I)
      IF(IQML.EQ.0) GOTO 2005
      DO 2006 JI=1,ISG(ICC)
      IAX=IABI(JJ,JI,1,ICC)
      IBX=IABI(JJ,JI,2,ICC)
      IF(IAX.EQ.KAX.AND.IBX.EQ.KBX) THEN
      KKK=KKK+1
      IF(KKK.GT.40) THEN
      PRINT *,'KKK OVER 40'
      STOP
      ENDIF
C     PRINT *,'J (MODEL)=',J,' JI (SCENE)=',JI,' KKK=',KKK
```

```
      NFN(I,K,KKK,2)=JI
      NFN(I,K,KKK,1)=J
      ENDIF
 2006 CONTINUE
 2005 CONTINUE
 2002 CONTINUE
 2000 CONTINUE
C
      RETURN
      END
C
      SUBROUTINE CHILD(CFN,NFN,MBL,IOUT,ICAN,MMS)
      INTEGER*2 CFN(11,40,2),NFN(11,40,40,2)
      DIMENSION P(40),NP(40),LP(40),KILL(3)
      INTEGER*2 MM(40,30),MMS(3,11,3,30)
      WRITE(IOUT,1000)
 1000 FORMAT(1H1)
      DO 1 I=1,11
      DO 1 J=1,3
      DO 1 K=1,30
    1 MMS(ICAN,I,J,K)=0
C
      DO 2011 I=1,MBL
CCX   WRITE(IOUT,1001)
CCX   WRITE(IOUT,1001)
CCX   WRITE(IOUT,1002) I
 1002 FORMAT(1H ,'MODEL NUMBER=',I2)
CCX   WRITE(IOUT,1001)
 1001 FORMAT(1H )
      DO 2010 K=4,40
      IF(CFN(I,K,1).EQ.0) GOTO 2010
CCX   WRITE(IOUT,3002)
 3002 FORMAT(1H )
      KKK1=NFN(I,K,40,1)
      KKK2=NFN(I,K,40,2)
CCX      WRITE(IOUT,3000)KKK1,(NFN(I,K,KK,1),KK=1,30)
CCX      WRITE(IOUT,3001)KKK2,(NFN(I,K,KK,2),KK=1,30)
 3000 FORMAT(1H ,'BROTHER OF FATHER (MODEL)=',I2,5X,'CHILDREN',4X,30I2)
 3001 FORMAT(1H ,'PAIR',14X,'(SCENE)=',I2,5X,'PAIR',5X,'=',2X,30I2)
 2010 CONTINUE
 2011 CONTINUE
C
C
      DO 100 I=1,MBL
C
```

```
      DO 98 I1=1,40
      DO 98 I2=1,30
      MM(I1,I2)=0
   98 CONTINUE
C
      DO 99 II=1,40
      NP(II)=0
   99 P(II)=0.0
C
CC    PRINT *,'ICAN=',ICAN
      KK3=0
      DO 101 K=4,40
      IF(CFN(I,K,1).EQ.0) GOTO 101
      KK3=KK3+1
      KK4=1
      MM(KK3,1)=NFN(I,K,1,1)
CC    PRINT *,'KK3=',KK3,' KK4=',KK4,' MM(KK3,1)=',MM(KK3,1)
      DO 102 KK=2,30
      IF((NFN(I,K,KK,1).NE.0).AND.
     &(NFN(I,K,KK-1,1).NE.NFN(I,K,KK,1))) THEN
      KK4=KK4+1
      MM(KK3,KK4)=NFN(I,K,KK,1)
CC    PRINT *,'KK3=',KK3,' KK4=',KK4,' MM(KK3,KK4)=',MM(KK3,KK4)
      ENDIF
  102 CONTINUE
      P(KK3)=-FLOAT(KK4)
  101 CONTINUE
CC    PRINT *,'KK3=',KK3
      CALL XORDER(P,NP,KK3,KK3)
      DO 1029 IS=1,3
 1029 KILL(IS)=0
      KK5=MINO(3,KK3)
      IAIA=0
      DO 103 IAQ=1,KK5
      IAIA=IAIA+1
 1030 CONTINUE
      DO 104 IS=1,KK3
      IF(NP(IS).EQ.IAIA) GOTO 1606
  104 CONTINUE
      PRINT *,'IAIA ERROR'
      STOP
 1606 CONTINUE
C
CC    PRINT *,'KILL(IKK)=',(KILL(IKK),IKK=1,3)
C
```

```
      DO 1919 IKK=1,KK5
 1919 IF(NFN(I,IS+3,40,1).EQ.KILL(IKK)) GOTO 1677
      GOTO 1607
 1677 CONTINUE
      IAIA=IAIA+1
CCC   PRINT *,'IAIA=',IAIA
      GOTO 1030
 1607 CONTINUE
      KILL(IAQ)=NFN(I,IS+3,40,1)
CC    PRINT *,'IAQ=',IAQ,' IS=',IS
CCCCCCC
CCC       PRINT *,'ICAN=',ICAN
C     MODEL CATEGORY
      MMS(ICAN,I,IAQ,1)=I
C     SCENE CATEGORY
      MMS(ICAN,I,IAQ,2)=CFN(I,1,2)
C     FATHER FACET-1
      MMS(ICAN,I,IAQ,3)=CFN(I,2,1)
C     FATHER FACET-2
      MMS(ICAN,I,IAQ,4)=CFN(I,2,2)
C     DISTANCE
      MMS(ICAN,I,IAQ,5)=CFN(I,3,1)
C     A BROTHER FACET-1
      MMS(ICAN,I,IAQ,6)=NFN(I,IS+3,40,1)
C     A BROTHER FACET-2
      MMS(ICAN,I,IAQ,7)=NFN(I,IS+3,40,2)
CCCCC
      DO 106 KK=1,20
  106 MMS(ICAN,I,IAQ,KK+7)=MM(IS,KK)
C
CCX   WRITE(IOUT,199) ICAN,I,IAQ,(MMS(ICAN,I,IAQ,IK),IK=1,20)
  199 FORMAT(1H ,'ICAN=',1X,I2,1X,'I=',I2,' IAQ=',I2,
     &' MCAT=',I2,' SCAT=',I2,' FFCT=',2I2,' DIS=',I4,' BFCT=',2I2,1X
     &,'FACET=',13I2)
C
  103 CONTINUE
  100 CONTINUE
      RETURN
      END
C
      SUBROUTINE XORDER(XY,NXY,M,MM)
      DIMENSION XY(1),NXY(1),SD(1000)
      SD(1)=10000.0
      DO 900 K=1,M
      NXY(K)=10000
```

```
  900 SD(1)=AMIN1(SD(1),XY(K))
      DO 901 K=1,M
  901 IF(ABS(SD(1)-XY(K)).LT.0.000001) GOTO 1
    1 NXY(K)=1
      IF(M.EQ.1) GOTO 904
      DO 902 KK=2,MM
      SD(KK)=10000.0
      DO 903 IJ=1,M
      IF(NXY(IJ).LE.KK-1) GOTO 903
      SD(KK)=AMIN1(SD(KK),XY(IJ))
  903 CONTINUE
      DO 906 IL=1,M
      IF(NXY(IL).LE.KK-1) GOTO 906
      IF(ABS(SD(KK)-XY(IL)).LT.0.00000001) GOTO 2
  906 CONTINUE
    2 NXY(IL)=KK
  902 CONTINUE
  904 CONTINUE
      RETURN
      END
C
      SUBROUTINE OUT1(IOUT,MBL,MSG,MSP,ATT)
      DIMENSION MSG(11),MSP(20,11),ATT(111,9)
      WRITE(IOUT,100)
      DO 1 MB=1,MBL
      WRITE(IOUT,100)
      MMM=MSG(MB)
      DO 2 I=1,MMM
      IS=MSP(I,MB)
      IX=IFIX(ATT(IS,1)+0.5)
      IY=IFIX(ATT(IS,2)+0.5)
      IZ=IFIX(ATT(IS,3))
      NX=IFIX(ATT(IS,4)*100.0)
      NY=IFIX(ATT(IS,5)*100.0)
      NZ=IFIX(ATT(IS,6)*100.0)
      IAR=IFIX(ATT(IS,7))
CCX   WRITE(IOUT,101) MB,I,IS,IX,IY,IZ,NX,NY,NZ,IAR
    2 CONTINUE
    1 CONTINUE
  100 FORMAT(1H )
  101 FORMAT(1H ,'BLOB=',I2,1X,'LFCT=',I2,1X,'GFCT=',I2,1X,
     &'IX= ',I3,1X,'IY= ',I3,1X,'IZ= ',I3,'  NX=',I3,' NY=',I3,' NZ='
     &,I3,' AREA=',I5)
      RETURN
      END
```

```
C
      SUBROUTINE OUT2(IOUT,MBL,MSG,MABM)
      DIMENSION MSG(11)
      INTEGER*2 MABM(20,20,2,11)
      WRITE(IOUT,101)
      DO 1 MB=1,MBL
      MMM=MSG(MB)
      DO 2 I=1,MMM
CCX   WRITE(IOUT,100) MB,I,((MABM(I,J,IK,MB),IK=1,2),J=1,MMM)
    2 CONTINUE
CCX   WRITE(IOUT,101)
    1 CONTINUE
  100 FORMAT(1H ,'BLOB NO= ',I2,1X,'LOC FCT= ',I2,1X,15('('
     &,I2,',',I2,')'))
  101 FORMAT(1H )
      RETURN
      END
C
C
      SUBROUTINE DELTA(NX,NY,NZ,THETA,PHAI)
      XL=FLOAT(NX)
      XM=FLOAT(NY)
      XN=FLOAT(NZ)
      IF(SQRT(XM**2+XL**2).LE.6.0) THEN
      THETA=0.0
      ELSE
      IF(ABS(XL).LE.0.00000001) THEN
      THETA=0.0
      ELSE
      THETA=ATAN2(XL,XM)
      ENDIF
      ENDIF
C
      XML=SQRT(XM**2+XL**2)
      IF(SQRT(XML**2+XN**2).LE.6.0) THEN
      PHAI=0.0
      ELSE
      IF(ABS(XN).LE.0.00000001) THEN
      PHAI=0.0
      ELSE
      PHAI=ATAN2(XN,XML)
      ENDIF
      ENDIF
C
      RETURN
```

```
      END
C
C
      SUBROUTINE ROTATE(XI,YI,ZI,THETA,PHAI,XX,YY,ZZ)
      XX=COS(PHAI)*COS(THETA)*XI+COS(PHAI)*SIN(THETA)*YI
     &   -SIN(PHAI)*ZI
      YY=-SIN(THETA)*XI+COS(THETA)*YI
      ZZ=SIN(PHAI)*COS(THETA)*XI+SIN(PHAI)*SIN(THETA)*YI
     &   +COS(PHAI)*ZI
C
      RETURN
      END
C
C
      SUBROUTINE MODEL(MDL,ATT,MSP,MSG,AMM,BTT,ISP,ISG,BMM,IOUT)
      DIMENSION MDL(9,4,2),ATT(111,9),MSP(20,11),MSG(11)
      DIMENSION           BTT(111,9),ISP(20,11),ISG(11)
      DIMENSION AMM(20,11),BMM(20,11)
      DIMENSION           CTT(111,9),ICSP(20,11),ICSG(11),CMM(20,11)
      DIMENSION SK(7),K2K(20)
C
C  IG..... GLOBAL FACET NUMBER
C  I ..... LOCAL FACET NUMBER
C  MB..... BLOB NUMBER
C
C  ATT,BTT.... ATT(IG,.)
C  MSG,ISG.... MSG(MB).. TOTAL FACET NUMBER OF EACH BLOB
C  MSP,ISP.... MSP(I,MB) GLOBAL FACET NUMBER OF I-TH FACET IN BLOB MB
C  AMM,BMM.... AMM(I,MB) AREA OF BLOB NUMBER MB
C
      DO 1000 I=1,9
      IF(MDL(I,1,1).EQ.0) GOTO 1000
      MBL=MDL(I,1,1)
      MLN=MDL(I,1,2)
      IBL=MDL(I,2,1)
      ILN=MDL(I,2,2)
C
      NN=MSP(MLN,MBL)
      A1=ATT(NN,1)
      A2=ATT(NN,2)
      A3=ATT(NN,3)
C
      DO 1 J=1,MSG(MBL)
      NQ=MSP(J,MBL)
      ATT(NQ,1)=ATT(NQ,1)-A1
```

```
      ATT(NQ,2)=ATT(NQ,2)-A2
      ATT(NQ,3)=ATT(NQ,3)-A3
    1 CONTINUE
C
      MM=ISP(ILN,IBL)
      B1=BTT(MM,1)
      B2=BTT(MM,2)
      B3=BTT(MM,3)
C
      DO 2 J=1,ISG(IBL)
      NQ=ISP(J,IBL)
      BTT(NQ,1)=BTT(NQ,1)-B1
      BTT(NQ,2)=BTT(NQ,2)-B2
      BTT(NQ,3)=BTT(NQ,3)-B3
    2 CONTINUE
C
      NX=IFIX(ATT(NN,4)*100.0)
      NY=IFIX(ATT(NN,5)*100.0)
      NZ=IFIX(ATT(NN,6)*100.0)
C
      CALL DELTA(NX,NY,NZ,THETA1,PHAI1)
      PRINT *,'NX=',NX,'  NY=',NY,'  NZ=',NZ
      PRINT *,'THETA1',THETA1,'  PHAI1=',PHAI1
      WRITE(IOUT,6000) NX,NY,NZ,THETA1,PHAI1
 6000 FORMAT(1H ,'NX,NY,NZ=',3I4,'THETA1=',F5.2,' PHAI1=',F5.2)
C
      NX=IFIX(BTT(MM,4)*100.0)
      NY=IFIX(BTT(MM,5)*100.0)
      NZ=IFIX(BTT(MM,6)*100.0)
C
      CALL DELTA(NX,NY,NZ,THETA2,PHAI2)
      PRINT *,'NX=',NX,'  NY=',NY,'  NZ=',NZ
      PRINT *,'THETA2=',THETA2,'  PHAI2=',PHAI2
      WRITE(IOUT,6001) NX,NY,NZ,THETA2,PHAI2
 6001 FORMAT(1H ,'NX,NY,NZ=',3I4,'THETA2=',F5.2,' PHAI2=',F5.2)
C
      THETA=THETA2-THETA1
      PHAI=PHAI2-PHAI1
      PRINT *,'THETA=',THETA,'  PHAI=',PHAI
      WRITE(IOUT,6002) THETA,PHAI
 6002 FORMAT(1H ,'THETA=',F5.2,'  PHAI=',F5.2)
C
      DO 100 L=1,ISG(IBL)
C
      NN=ISP(L,IBL)
```

```
      NX=IFIX(BTT(NN,4)*100.0)
      NY=IFIX(BTT(NN,5)*100.0)
      NZ=IFIX(BTT(NN,6)*100.0)
      CALL DELTA(NX,NY,NZ,THETA3,PHAI3)
      PRINT *,'NX=',NX,'  NY=',NY,'  NZ=',NZ
      PRINT *,'THETA3=',THETA3,'  PHAI3=',PHAI3
      WRITE(IOUT,6003) NX,NY,NZ,THETA3,PHAI3
 6003 FORMAT(1H ,'NX,NY,NZ=',3I4,' THETA3=',F5.2,'  PHAI3=',F5.2)
      THETAX=THETA3+THETA
      PHAIX=PHAI3+PHAI
      PRINT *,'THETAX=',THETAX,'  PHAIX=',PHAIX
      WRITE(IOUT,6004) THETAX,PHAIX
 6004 FORMAT(1H ,'THETAX=',F5.2,'  PHAIX=',F5.2)
      BTT(NN,4)=COS(PHAIX)*SIN(THETAX)
      BTT(NN,5)=COS(PHAIX)*COS(THETAX)
      BTT(NN,6)=SIN(PHAIX)
      XI=BTT(NN,1)
      YI=BTT(NN,2)
      ZI=BTT(NN,3)
      CALL ROTATE(XI,YI,ZI,THETA,PHAI,XX,YY,ZZ)
      WRITE(IOUT,6010) XI,YI,ZI,XX,YY,ZZ
 6010 FORMAT(1H ,'XI,YI,ZI=',3(F7.2,1X),'  XX,YY,ZZ=',3(F7.2,1X))
      BTT(NN,1)=XX
      BTT(NN,2)=YY
      BTT(NN,3)=ZZ
C
  100 CONTINUE
 1000 CONTINUE
C
C*** MERGING
C
      ISCENE1=1
      ISCENE2=2
C
      DO 2000 I=1,9
      IF(MDL(I,1,1).EQ.0) GOTO 2000
      MBL=MDL(I,1,1)
      MLN=MDL(I,1,2)
      IBL=MDL(I,2,1)
      ILN=MDL(I,2,2)
      WRITE(IOUT,4000) ISCENE1,MBL,MLN
      DO 2001 J=1,MSG(MBL)
      NN=MSP(J,MBL)
 2001 WRITE(IOUT,4001) I,J,NN,(ATT(NN,K),K=1,7)
 4000 FORMAT(1H ,'SCENE NO=',I2,' BLOB NO=',I2,' FACET NO=',I2)
```

```
 4001 FORMAT(1H ,'BLOB NUM=',I2,' LOCAL FACET NO=',I2,' GLOBAL
     &FACET NO=',I2,1X,7(F7.1,1X))
      WRITE(IOUT,4002)
 4002 FORMAT(1H )
C
      WRITE(IOUT,4000) ISCENE2,IBL,ILN
      DO 2002 J=1,ISG(IBL)
      NN=ISP(J,IBL)
 2002 WRITE(IOUT,4001) I,J,NN,(BTT(NN,K),K=1,7)
C
      MBLZ=MDL(I,3,1)
      MLNZ=MDL(I,3,2)
      IBLZ=MDL(I,4,1)
      ILNZ=MDL(I,4,2)
      NN=MSP(MLNZ,MBLZ)
      MM=ISP(ILNZ,IBLZ)
      NX=IFIX(ATT(NN,1)+0.5)
      NY=IFIX(ATT(NN,2)+0.5)
      NZ=IFIX(ATT(NN,3)+0.5)
      CALL DELTA(NX,NY,NZ,THETA4,PHAI4)
      WRITE(IOUT,6200) THETA4,PHAI4
 6200 FORMAT(1H ,'THETA4=',F7.2,'  PHAI4=',F7.2)
      NX=IFIX(BTT(MM,1)+0.5)
      NY=IFIX(BTT(MM,2)+0.5)
      NZ=IFIX(BTT(MM,3)+0.5)
      CALL DELTA(NX,NY,NZ,THETA5,PHAI5)
      WRITE(IOUT,6201) THETA5,PHAI5
 6201 FORMAT(1H ,'THETA5=',F7.2,'  PHAI5=',F7.2)
      THETA6=THETA4-THETA5
      PHAI6=PHAI4-PHAI5
      WRITE(IOUT,6202) THETA6,PHAI6
 6202 FORMAT(1H ,'THETA6=',F7.2,'  PHAI6=',F7.2)
C
      DO 200 L=1,ISG(IBLZ)
C
      NN=ISP(L,IBLZ)
      NX=IFIX(BTT(NN,4)*100.0)
      NY=IFIX(BTT(NN,5)*100.0)
      NZ=IFIX(BTT(NN,6)*100.0)
      CALL DELTA(NX,NY,NZ,THETA7,PHAI7)
      PRINT *,'NX=',NX,'  NY=',NY,'  NZ=',NZ
      PRINT *,'THETA7=',THETA7,'  PHAI7=',PHAI7
      WRITE(IOUT,6103) NX,NY,NZ,THETA7,PHAI7
 6103 FORMAT(1H ,'NX,NY,NZ=',3I4,' THETA7=',F5.2,'  PHAI7=',F5.2)
      THETAX=THETA7+THETA6
```

```
      PHAIX=PHAI7+PHAI6
      PRINT *,'THETAX=',THETAX,'  PHAIX=',PHAIX
      WRITE(IOUT,6104) THETAX,PHAIX
 6104 FORMAT(1H ,'THETAX=',F5.2,'  PHAIX=',F5.2)
      BTT(NN,4)=COS(PHAIX)*SIN(THETAX)
      BTT(NN,5)=COS(PHAIX)*COS(THETAX)
      BTT(NN,6)=SIN(PHAIX)
      XI=BTT(NN,1)
      YI=BTT(NN,2)
      ZI=BTT(NN,3)
      CALL ROTATE(XI,YI,ZI,THETA6,PHAI6,XX,YY,ZZ)
      WRITE(IOUT,6110) XI,YI,ZI,XX,YY,ZZ
 6110 FORMAT(1H ,'XI,YI,ZI=',3(F7.2,1X),'  XX,YY,ZZ=',3(F7.2,1X))
      BTT(NN,1)=XX
      BTT(NN,2)=YY
      BTT(NN,3)=ZZ
C
  200 CONTINUE
      WRITE(IOUT,4000) ISCENE1,MBL,MLN
      DO 2101 J=1,MSG(MBL)
      NN=MSP(J,MBL)
 2101 WRITE(IOUT,4001) I,J,NN,(ATT(NN,K),K=1,7)
      WRITE(IOUT,4002)
C
      WRITE(IOUT,4000) ISCENE2,IBL,ILN
      DO 2102 J=1,ISG(IBL)
      NN=ISP(J,IBL)
 2102 WRITE(IOUT,4001) I,J,NN,(BTT(NN,K),K=1,7)
C
C
 2000 CONTINUE
C
      IG=0
      DO 3000 I=1,9
      ICSG(I)=0
      IF(MDL(I,1,1).EQ.0) THEN
      WRITE(IOUT,5880) I
 5880 FORMAT(1H ,'MDL(I,1,1)=0, I=',I2)
      DO 3001 K=1,MSG(I)
      IG=IG+1
      MM=MSP(K,I)
      ICSP(K,I)=IG
      DO 3002 L=1,7
 3002 CTT(IG,L)=ATT(MM,L)
 3001 CONTINUE
```

```
      ICSG(I)=MSG(I)
      ELSE
C
      IC=0
      KK=MDL(I,1,1)
      LL=MDL(I,2,1)
      DO 305 K2=1,ISG(LL)
  305 K2K(K2)=1
C
C
      DO 3004 K1=1,MSG(KK)
      DO 3014 L=1,7
 3014 SK(L)=0.0
      IGGG=0
      J1=MSP(K1,KK)
CC
CC
      DO 3005 K2=1,ISG(LL)
      J2=ISP(K2,LL)
      SS1=0.0
      DO 3006 L=1,3
 3006 SS1=SS2+SQRT((ATT(J1,L)-BTT(J2,L))**2)
      SS2=0.0
      DO 3007 L=4,6
 3007 SS2=SS2+SQRT((ATT(J1,L)-BTT(J2,L))**2)
      IF(SS1.LE.5.0.AND.SS2.LE.0.2) THEN
      K2K(K2)=0
      WRITE(IOUT,5454) K1,K2
 5454 FORMAT(1H ,'MERGE BY SS1 SS2 K1=',I2,'K2=',I2)
      IGGG=IGGG+1
      DO 3008 L=1,7
 3008 SK(L)=SK(L)+BTT(J2,L)
      ENDIF
 3005 CONTINUE
CC
CC
      WRITE(IOUT,5890) K1,IGGG
 5890 FORMAT(1H ,'K1=',I3,'  IGGG=',I3)
      IF(IGGG.NE.0) THEN
      IC=IC+1
      IG=IG+1
      ICSP(IC,I)=IG
      WRITE(IOUT,5900) IC,IG
 5900 FORMAT(1H ,'IC= ',I3,' IG= ',I3)
      IF(IG.GE.111) STOP
```

```
      DO 3009 L=1,7
 3009 CTT(IG,L)=(ATT(J1,L)+SK(L)/FLOAT(IGGG))/2.0
      ELSE
      IC=IC+1
      IG=IG+1
      ICSP(IC,I)=IG
      WRITE(IOUT,5877) IC,IG
 5877 FORMAT(1H ,' IGGG=0, IC=',I3,' IG=',I3)
      DO 3200 L=1,7
 3200 CTT(IG,L)=ATT(J1,L)
      ENDIF
C
 3004 CONTINUE
CC
CC
      DO 3055 K2=1,ISG(LL)
      IF(K2K(K2).EQ.1) THEN
      IC=IC+1
      IG=IG+1
      WRITE(IOUT,5998) K2,IC,IG
 5998 FORMAT(1H ,'K2K(K2)=0,K2= ',I3,' IC= ',I3,' IG= ',I3)
      IF(IG.GE.111) STOP
      ICSP(IC,I)=IG
      DO 3056 L=1,7
 3056 CTT(IG,L)=BTT(ISP(K2,LL),L)
      ENDIF
 3055 CONTINUE
C
      ENDIF
      ICSG(I)=IC
 3000 CONTINUE
C
      DO 3500 I=1,9
      IF(MDL(I,1,1).EQ.0) GOTO 3500
      WRITE(IOUT,9330)I,ICSG(I)
 9330 FORMAT(1H ,'I=',I2,2X,'ICSG(I)=', I2)
      DO 3501 J=1,ICSG(I)
      JJ=ICSP(J,I)
 3501 WRITE(IOUT,4900) I,JJ,(CTT(JJ,L),L=1,7)
 3500 CONTINUE
 4900 FORMAT(1H ,'I=',I2,'JJ=',I3,'  CTT=',7(F6.2,1X))
      RETURN
      END
C
C
```

```
C
      SUBROUTINE FINAL(MBL,ATT,BTT,MSP,ISP,MSG,MMS,IOUT)
      INTEGER MSG(11),LIKE(3,11)
      INTEGER*2 MMS(3,11,3,30)
      DIMENSION XLIK(3,11),XX(3),NX(3)
      DIMENSION ATT(111,9),BTT(111,9),MSP(20,11),ISP(20,11)
C
      DO 1 I=1,MBL
      DO 2 ICAN=1,3
      LIKE(ICAN,I)=0
      DO 3 J1=8,27
      M1=MMS(ICAN,I,1,J1)
      IF(M1.EQ.0) GOTO 3
      DO 4 J2=8,27
      M2=MMS(ICAN,I,2,J2)
      IF(M1.EQ.M2) THEN
      DO 5 J3=8,27
      M3=MMS(ICAN,I,3,J3)
    5 IF(M1.EQ.M3) LIKE(ICAN,I)=LIKE(ICAN,I)+1
      ENDIF
    4 CONTINUE
    3 CONTINUE
      XLIK(ICAN,I)=FLOAT(LIKE(ICAN,I))/FLOAT(MSG(I))
    2 CONTINUE
    1 CONTINUE
C
C
C
      DO 10 I=1,MBL
CCX   WRITE(IOUT,1000)
 1000 FORMAT(1H )
C     WRITE(IOUT,1001)
C1001 FORMAT(1H ,'MODEL NO. (FATHER FACET)  SCENE NO. (FATHER FACET)')
CC    DO 11 ICAN=1,3
CC 11 WRITE(IOUT,1001) I,ICAN, (MMS(ICAN,I,1,J),J=1,7),XLIK(ICAN,I)
 1001 FORMAT(1H ,9(I4),F6.2)
   10 CONTINUE
C
C
      DO 300 I=1,MBL
      DO 200 ICAN=1,3
      XX(ICAN)=-XLIK(ICAN,I)
  200 CONTINUE
C
      CALL XORDER(XX,NX,3,3)
```

```
C
      DO 201 ICAN=1,3
      IF(NX(ICAN).EQ.1) IA1=ICAN
      IF(NX(ICAN).EQ.2) IA2=ICAN
  201 CONTINUE
CC    PRINT *,'I=',I,' IA1=',IA1,' IA2=',IA2
C
      IRSLT=IA1
      IF(MMS(IA1,I,1,5).GE.500) IRSLT=0
C
      IF((-XX(IA2).GE.0.7).AND.
```